



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

DISEÑO Y EVALUACIÓN DE ARQUITECTURA DE SOLUCIÓN DE PREVENCIÓN
ACTIVA PARA CICLISTAS

TESIS PARA OPTAR AL GRADO DE
MAGÍSTER EN CIENCIAS, MENCIÓN COMPUTACIÓN

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN

SALOMÓN ALEJANDRO TORRES SILVA

PROFESORA GUÍA:
SANDRA CÉSPEDES UMAÑA

PROFESOR CO-GUÍA:
JAVIER BUSTOS JIMÉNEZ

MIEMBROS DE LA COMISIÓN:
JOSÉ PINO URTUBIA
LUIS MATEU BRULÉ
MANUEL VILLALOBOS CID

SANTIAGO DE CHILE
2022

RESUMEN DE LA TESIS PARA OPTAR
AL TÍTULO DE MAGÍSTER EN CIENCIAS, MENCIÓN COMPUTACIÓN
POR: SALOMÓN ALEJANDRO TORRES SILVA
FECHA: 2022
PROF. GUÍA: SANDRA CÉSPEDES UMAÑA
PROF. CO-GUÍA: JAVIER BUSTOS JIMÉNEZ

DISEÑO Y EVALUACIÓN DE ARQUITECTURA DE SOLUCIÓN DE PREVENCIÓN ACTIVA PARA CICLISTAS

Los ciclistas cuentan con pocos métodos de prevención activa de accidentes. En especial en intersecciones de calle, los ciclistas están expuestos a las imprudencias de los automovilistas que pueden ocasionar accidentes, siendo los ciclistas los que sufren los daños mas severos en caso de haber una colisión.

En este trabajo de tesis, se propone diseñar y estudiar una arquitectura de solución de prevención activa para ciclistas. Esta arquitectura de solución consiste en tomar la información de posición y velocidad del automóvil, enviarlo a una unidad de procesamiento y finalmente enviar una alerta al ciclista si es que hay peligro inminente de colisión.

Para la construcción de esta arquitectura de solución, se toma un algoritmo de predicción de giros vehiculares, y se adapta para que se procese en línea consumiendo los datos de entrada por medio de una red de comunicación. El algoritmo recibe los datos del vehículo a analizar y posteriormente entrega una alerta si es que es probable que el vehículo realice un giro hacia la derecha en la siguiente intersección de calle.

La otra parte de la arquitectura de solución consiste en construir una arquitectura de comunicación utilizando el protocolo IEEE 802.11p junto con otros protocolos para entregar los mensajes de alerta al ciclista. Para la arquitectura de solución, se estudian diferentes arquitecturas y protocolos para determinar el rendimiento de cada configuración.

La solución, con las diferentes combinaciones de arquitecturas a estudiar, se construyen en un ambiente simulado utilizando herramientas de simulación como son SUMO, OMNET++ y Veins. Con estas herramientas se integra el algoritmo de predicción de giros en conjunto con las arquitecturas a estudiar, en un escenario donde se puede probar la arquitectura de solución completa.

Para comprobar la efectividad y utilidad de la arquitectura de solución propuesta, se utilizan dos métricas: *Time to Collision (TTC)* y *Precisión de la predicción*. Estas métricas se encargan de medir si es que la arquitectura de solución entrega la alerta con tiempo suficiente y la otra métrica se encarga de ver que tan efectivo es el algoritmo de predicción de giros para inferir el comportamiento de los automovilistas.

La arquitectura de solución diseñada propone una nueva alternativa de prevención activa para los ciclistas. Los resultados obtenidos muestran la efectividad de la arquitectura de solución para entregar alertas a un ciclista permitiéndole realizar maniobras evasivas para prevenir o mitigar un accidente en escenarios específicos.

Agradecimientos

En primer lugar, quiero agradecer a Dios por todo lo que me ha dado. Si no fuera por su ayuda, no hubiese podido llegar a este punto.

Agradezco a Luvino mi padre, que me inculcó el esfuerzo, la dedicación y la motivación para seguir adelante sin importar las dificultades para la vida. Un abrazo grande para el cielo, donde espero encontrarlo un día. Infinitas gracias Gemima mi madre, que siempre estuvo a mi lado instándome, alentándome y entregándome el cariño necesario para los días difíciles. Gracias a mis hermanos y amigos, que siempre estuvieron presentes en esta larga jornada y me ayudaron a levantarme cada vez que no tenía ganas de continuar.

Muchísimas gracias a mi novia, que sin su ayuda y constante preocupación, hubiera olvidado concluir este proceso.

Finalmente, muchas gracias a Sandra y Javier, que tuvieron la paciencia y dedicación para apoyarme, asesorarme y corregirme durante estos largos años.

A todos, infinitas gracias.

Los quiero mucho.

Tabla de Contenido

Introducción	1
1. Motivación y Antecedentes	3
1.1. Contexto	3
1.2. Problema a Resolver	4
1.3. Hipótesis	4
1.4. Objetivos	4
1.4.1. Objetivo General	4
1.4.2. Objetivos específicos	4
1.5. Contribuciones	5
1.6. Metodología	5
2. Estado del Arte	7
2.1. Sistema de transporte inteligente	7
2.2. Algoritmo de predicción	7
2.2.1. Variables físicas	7
2.2.2. Maniobras del conductor	8
2.2.3. Interacciones e interdependencias de agentes de tráfico	8
2.3. Arquitecturas de Comunicación	9
2.3.1. Redes vehiculares	9
2.3.2. Sistemas de comunicación	9
2.4. Aplicaciones de alarma para ciclistas	10
2.5. Resumen	10
3. Modelo de Predicción	12
3.1. Descripción del algoritmo de giro	12
3.2. Adaptación para procesamiento en línea	15
3.2.1. Complejidad espacial	16
3.2.2. Complejidad temporal	16
3.3. Adaptaciones de implementación del algoritmo	17
4. Arquitectura de Comunicación	19
4.1. VANET	20
4.1.1. WAVE	20
4.1.2. IEEE 802.11p	20
4.1.3. IEEE 1609	22
4.1.4. LTE	24

4.1.5. Métricas de evaluación	24
5. Evaluación de arquitecturas de comunicación	28
5.1. Origen de los Datos	28
5.1.1. TAPAS Cologne Scenario	28
5.1.2. Formato de los Datos	29
5.2. Procesamiento de los Datos	29
5.3. Resultados	32
5.3.1. Descripción de la intersección	33
5.4. Vehículo - Bicicleta	33
5.4.1. Time To Collision	34
5.4.2. Precisión de la Predicción	36
5.5. Vehículo - RSU - Bicicleta	37
5.5.1. Time To Collision	37
5.5.2. Precisión de la Predicción	40
5.6. Vehículo - RSU - LTE - Bicicleta	41
5.6.1. Time To Collision	41
5.6.2. Precisión de la Predicción	43
5.7. Vehículo - RSU - Server - Bicicleta	45
5.7.1. Time To Collision	46
5.7.2. Precisión de la Predicción	47
5.8. Precisión según cantidad de vehículos	49
5.9. Discusión Resultados	51
5.9.1. Time To Collision	51
5.9.2. Velocidad Máxima	52
5.9.3. Pérdida de Paquetes	53
Conclusión	54
5.10. Trabajo Futuro	56
5.10.1. Arquitecturas con 5G	56
5.10.2. Otros Algoritmos de predicción	56
5.10.3. Métodos de alerta para ciclistas	57
Bibliografía	58

Índice de Tablas

2.1. Algoritmos de predicción	10
2.2. Arquitecturas de comunicación	11
3.1. Hipótesis de inferencia de intención del conductor	13
3.2. Parámetros del Modelo de Conductor Inteligente de acuerdo a [23]	13
4.1. Parámetros de IEEE 802.11P. Basada en [41]	21
4.2. Criterios clave de rendimiento. Basada en [28]	25
4.3. Tiempo total requerido por un ciclista para frenar completamente según la velocidad, considerando tasa de desaceleración igual a $4.8 m/s^2$	25
4.4. Matriz de confusión para giros hacia la derecha	26
5.1. Descripción de atributos de un viaje	29
5.2. Parámetros de ejecución y sus descripciones	29
5.3. Atributos asociados a rutas con aristas incluidas	30
5.4. Atributos asociados a rutas con aristas finales	30
5.5. Parámetros de ejecución y sus descripciones	31
5.6. Descripción de los datos contenidos en archivo FILE.xml	31
5.7. Estructura de los datos resultantes al ejecutar traceExporter.py	32
5.8. Estructura de datos procesados por get_specific_gps_traces.py	32
5.9. Arquitecturas de comunicación propuestas y sus puntos de procesamiento	32
5.10. Parámetros para el escenario vehículo bicicleta con diferentes velocidades	34
5.11. Parámetros para el escenario vehículo bicicleta para diferentes pérdida de paquetes	36
5.12. Parámetros para el escenario vehículo - RSU - bicicleta con diferentes velocidades	38
5.13. Parámetros para el escenario vehículo - RSU - bicicleta para diferentes pérdida de paquetes	40
5.14. Parámetros para el escenario vehículo - RSU - LTE - bicicleta con diferentes velocidades	42
5.15. Parámetros para el escenario vehículo - RSU - LTE - bicicleta para diferentes pérdida de paquetes	44
5.16. Parámetros para el escenario vehículo - RSU - Server - bicicleta con diferentes velocidades	46
5.17. Parámetros para el escenario vehículo - RSU - LTE - bicicleta para diferentes pérdida de paquetes	48
5.18. Precisión y resultados de predicción por cantidad de vehículos para 300 rutas	51
5.19. Precisión y resultados de predicción por cantidad de vehículos para 300 rutas	51

Índice de Ilustraciones

1.	Imagen del aumento de ciclistas en diferentes países entre los años 2017 y 2018. Esta imagen pertenece a eco-counter[10]	2
3.1.	Bayesian network	14
3.2.	Distribución de probabilidad para cada modelo M y parámetro de aceleración a dado que la parametrización fue obtenida por <i>clustering</i> (a) o desde el trazado de curvatura (b). Figura basada en [23]	15
4.1.	Arquitectura general propuesta de comunicación vehículo-ciclista	19
4.2.	Pila WAVE	21
5.1.	Intersección obtenida de Colonia, Alemania	33
5.2.	Arquitectura Vehículo-Ciclista	34
5.3.	Tiempo para colisión según velocidad máxima, para arquitectura Vehículo - Bicicleta	35
5.4.	TTC por cada porcentaje de pérdida de paquetes, para arquitectura Vehículo - Bicicleta	35
5.5.	Precisión por cada porcentaje de pérdida de paquetes, para arquitectura Vehículo - Bicicleta	37
5.6.	Arquitectura Vehículo - RSU - Ciclista	38
5.7.	Tiempo para colisión según velocidad máxima, para arquitectura Vehículo - RSU - Bicicleta	39
5.8.	TTC por cada porcentaje de pérdida de paquetes, para arquitectura Vehículo - RSU - Bicicleta	40
5.9.	Precisión por cada porcentaje de pérdida de paquetes, para arquitectura Vehículo - RSU - Bicicleta	41
5.10.	Arquitectura Vehículo - RSU - LTE - Ciclista	42
5.11.	Tiempo para colisión según velocidad máxima, para arquitectura Vehículo - RSU - LTE - Bicicleta	43
5.12.	Precisión por cada porcentaje de pérdida de paquetes, para arquitectura Vehículo - RSU - LTE - Bicicleta	44
5.13.	Cantidad de giros detectados por cada porcentaje de pérdida de paquetes, para arquitectura Vehículo - RSU - LTE - Bicicleta	45
5.14.	Arquitectura Vehículo - RSU - Server - Ciclista	46
5.15.	Tiempo para colisión según velocidad máxima, para arquitectura Vehículo - RSU - Servidor - Bicicleta	47

5.16. TTC por cada porcentaje de pérdida de paquetes, para arquitectura Vehículo - RSU - Servidor - Bicicleta	48
5.17. Cantidad de giros detectados por cada porcentaje de pérdida de paquetes, para arquitectura Vehículo - RSU - Servidor - Bicicleta	49
5.18. Escenario con cinco vehículos presentes en simultáneo en la vía	49
5.19. Precisión según la cantidad de vehículos	50
5.20. Tiempo para colisión según velocidad máxima, para las cuatro arquitecturas presentadas	52
5.21. TTC - velocidad máxima y TTC para bicicletas	53

Introducción

En Chile y el mundo, el uso de la bicicleta como medio de transporte ha aumentado en los últimos años. La empresa Eco-counter[10] publicó el índice mundial de ciclismo que muestra el crecimiento del uso de la bicicleta en diferentes partes del mundo, como se observa en la Figura 1. Considerando que hay mas ciclistas en las calles, mejorar la seguridad y prevención de accidentes para los ciclistas es fundamental.

Una forma de mejorar la seguridad y la prevención de accidentes, es otorgar a los ciclistas con herramientas de prevención activa, que les permitan evitar y mitigar las colisiones con automovilistas. Con estas herramientas, se podría disminuir la incidencia de muertes y lesiones para los ciclistas, mejorando la seguridad vial de estos.

En los últimos años se han desarrollado tecnologías de comunicación vehicular dentro del marco de las “Smart Cities” y del “Internet of Things”. Esto provee nuevas herramientas que se pueden utilizar para desarrollar arquitecturas de solución que alertan a los conductores de la presencia de otros usuarios vulnerables de las vías [9], como peatones, ciclistas o motociclistas, ayudando de esta forma a disminuir la cantidad de accidentes en las vías. Dentro de este campo de investigación, destacan las redes vehiculares que permiten la comunicación entre vehículos utilizando protocolos de comunicación diseñados específicamente para este propósito, como es el protocolo Wireless Access in Vehicular Enviroments (WAVE) que es el estándar IEEE 802.11p [11] el cual se está implementando actualmente en los vehículos para su uso dentro de las redes vehiculares.

Considerando lo anterior se propone una arquitectura de solución de alertas para ciclistas que les permita tomar medidas de prevención activa para evitar colisiones, utilizando las nuevas tecnologías de comunicación disponibles y la capacidad de los nuevos algoritmos y modelos para predecir giros vehiculares hacia la derecha.

Countries with the Largest Growth in Number of Bicycles Counted

Progression is calculated by comparing total bicycle counts between 2017 and 2018 in countries with at least 20 counters. This analysis is based on data coming from automated bicycle counters with at least 4 months of comparable and validated data.



Figura 1: Imagen del aumento de ciclistas en diferentes países entre los años 2017 y 2018. Esta imagen pertenece a eco-counter[10]

Capítulo 1

Motivación y Antecedentes

1.1. Contexto

Las colisiones entre automovilistas y ciclistas representan un gran porcentaje del total de accidentes de tránsito. Según datos publicado por la Comisión nacional de Seguridad de Tránsito (CONASET) [8] para Chile, el 6% del total de fallecidos en siniestros viales son ciclistas. Durante el 2016, en 3.953 accidentes de tránsito estuvo involucrada la bicicleta y en dichos accidentes resultaron 93 ciclistas fallecidos y 3.326 lesionados. Un punto particular de riesgo de colisión son las intersecciones debido a la frecuencia con que el flujo de automovilistas y ciclistas se cruzan, según señala el *Institute for Traffic Accident Research and Data Analysis* (ITARDA) [19]. Además, de acuerdo a “Traffic Safety Facts 2011”, *National Highway Traffic Safety Administration* (NHTSA) [25], en Estados Unidos un 31% de los accidentes de tránsito ocurrió en intersecciones.

El desarrollo de las comunicaciones vehiculares ha crecido en los últimos años, donde se ha considerado que este tipo de comunicaciones son fundamentales para mejorar la seguridad en las vías y mejorar el tráfico en general. Pero a pesar de eso según como se presentan las comunicaciones vehiculares en NHTSA [26], la inclusión de los ciclistas y peatones en las comunicaciones vehiculares se ve como algo a largo plazo y no a corto plazo e incluso, se presentan a los ciclistas como actores pasivos que informarían de su presencia a vehículos motorizados para prevenir colisiones. Es por esto, que el proveer una herramienta y uso de las comunicaciones vehiculares enfocada en los ciclistas, se hace necesario para demostrar empíricamente que los usuarios vulnerables de las vías (VRU por sus siglas en inglés) pueden ser usuarios activos de los métodos de prevención y mitigación de accidentes y aportar a mejorar la seguridad de las vías.

1.2. Problema a Resolver

Los ciclistas cuentan con muy pocos métodos de prevención activa que les permitan evitar o mitigar accidentes con vehículos. Un método de prevención activa para un ciclista, es poder recibir una alerta de un posible accidente para poder reaccionar preventivamente evitando o mitigando la colisión.

En especial, en intersecciones de calle los ciclistas quedan expuestos a las imprudencias y errores de los automovilistas, por lo que se debe poder construir un sistema que activamente permita a los ciclistas prevenir accidentes en intersecciones de calle. Para este sistema, es necesario un algoritmo de predicción de giros vehiculares hacia la derecha, que pueda generar alertas en caso de posibles giros. Además, se necesita diseñar y evaluar arquitecturas de comunicación que permitan enviar mensajes de alerta a un ciclista que pueda sufrir un accidente, según lo alertado por el algoritmo de predicción de giros.

1.3. Hipótesis

Utilizando el protocolo de comunicación IEEE 802.11p para construir una arquitectura de comunicación y un algoritmo de predicción de giros, es posible construir un sistema de alerta temprana para ciclistas, que alerte con tiempo suficiente a los ciclistas de potenciales accidentes.

1.4. Objetivos

1.4.1. Objetivo General

Diseñar, implementar y evaluar, a nivel de simulación, un sistema de alerta temprana a ciclistas con predicción de giros vehiculares para prevenir accidentes. Además, determinar las condiciones mínimas necesarias para que dicho sistema funcione considerando las métricas de *time to collision* y precisión de la predicción.

1.4.2. Objetivos específicos

- Estudiar el estado del arte relacionado a los algoritmos de predicción de giro, comunicaciones vehiculares y sistemas de alerta temprana para seleccionar el algoritmo de predicción y las posibles arquitecturas de comunicación del sistema a desarrollar.
- Implementar una adaptación del algoritmo de predicción de giro seleccionado en un entorno simulado para probar la métrica de precisión de predicción del algoritmo.
- Evaluar arquitecturas de comunicación para determinar las ventajas de cada una con respecto al desempeño del sistema de alerta y evaluar alternativas centralizadas y dis-

tribuidas de dicho sistema considerando *time to collision* y precisión de la predicción como métricas de desempeño.

- Probar las comunicaciones del sistema considerando: pérdida de paquetes, tiempo de procesamiento de los paquetes enviados, capacidad máxima de procesamiento de paquetes para determinar los límites de *time to collision* y precisión de la predicción del sistema.

1.5. Contribuciones

En esta tesis se diseña y evalúa un sistema de prevención activa para ciclistas, integrando un algoritmo de predicción de giros con diferentes arquitecturas de comunicación y evaluando su desempeño en base a métricas de tiempo para colisión y precisión de la predicción.

Específicamente, las contribuciones son las siguientes:

- Adaptación de un algoritmo de predicción de giros vehiculares hacia la derecha, para ser procesado en línea.
- Diseño e implementación de arquitecturas de comunicación utilizando el protocolo IEEE 802.11p
- Creación de escenarios de simulación, para integrar el algoritmo de predicción de giros con las diferentes arquitecturas de comunicación evaluadas.
- Evaluación en ambientes simulados, del sistema integrado de prevención activa para ciclistas, utilizando métricas de tiempo para colisión y precisión de la predicción.

1.6. Metodología

Se estudia el estado del arte para seleccionar un algoritmo de predicción de giros vehiculares hacia la derecha. El algoritmo seleccionado tiene la característica de utilizar la velocidad y posición como datos de entrada, considerando además diferentes perfiles de conducción que presentan los conductores de vehículos. Una vez seleccionado el algoritmo, se adapta para implementarse en los simuladores que se utilizan para estudiar el rendimiento y eficacia del algoritmo.

Para la construcción del simulador, se utilizará SUMO 1.2.0[34], OMNet++ 5.5.1[27] y el framework Veins 5.0[40]. SUMO provee el simulador microscópico de tráfico vehicular donde se muestra una intersección de calles y los vehículos que circulan por dicha intersección, incluyendo automóviles y bicicletas por sus ciclo-vías respectivas. Por otra parte, OMNet++ construye la red de comunicación entre los vehículos, proveyendo las capacidades necesarias para que cada vehículo de la red pueda comunicarse con otros vehículos presentes en la red, tales como las bicicletas. Finalmente, el framework Veins, realiza la función de coordinación entre el simulador de tráfico SUMO y el simulador de redes OMNet++. La implementación del algoritmo de predicción y las redes de comunicación se realizan utilizando el lenguaje de

programación C++.

Para validar el algoritmo de predicción de giro, se utilizarán los datos de Colonia, Alemania, donde se cuenta con la información del tráfico de 24 horas, donde la cantidad de registros es de un millón de viajes aproximadamente. Se utilizará la información de los viajes entre las 6:00 am y 8:00 am y se seleccionaran algunas calles donde se realizan giros hacia la derecha para probar el algoritmo. Una vez identificadas las calles se deben calcular los parámetros propuestos en el modelo del algoritmo en base a la curvatura de cada calle. En caso de que el algoritmo no entregue buenos resultados, se entrenará el algoritmo conforme los modelos que se proponen en [23] para mejorar dichos resultados y de esta forma lograr que el algoritmo cumpla las necesidades del sistema de alerta temprana.

Para evaluar las arquitecturas de comunicación, se describe los protocolos de comunicación involucrados. Para esto se presentan los protocolos de comunicación vehicular inalámbrica y además los protocolos de comunicación inalámbrica de banda ancha. Con estos protocolos se presentan las diferentes arquitecturas de comunicación a estudiar y se analiza su desempeño.

Para probar al sistema, se consideraran diferentes porcentajes de perdidas de paquetes entre el vehículo y la *Road-Side Unit (RSU)* , considerando la posibilidad de aumentar la perdida si es que las pruebas resultan exitosas. También se considerara cantidad variable de vehículos presentes en la intersección y ciclistas en la ciclo-vía. En todos los casos, se busca determinar el tiempo de predicción que otorga el sistema de predicción para el ciclista, desde el momento en que llega la alerta hasta que se produce el giro del vehículo y el porcentaje de éxito al predecir giros.

Capítulo 2

Estado del Arte

2.1. Sistema de transporte inteligente

Con el avance de los sistemas de transporte, las nuevas redes de comunicación y el aumento de ciclistas presentes en las vías públicas, se presenta en [29] que los ciclistas tiene varias necesidades que suplir cuando viajan por las calles, tales como seguridad, accesibilidad, comodidad, mas información y tiempos menores de traslado y todas estas necesidades pueden ser cubiertas en cierta medida con un sistema de transporte inteligente (ITS). Además, en [15] se presentan beneficios de contar con sistemas de transporte inteligentes para ciclistas, mejorando diferentes aspectos como la seguridad del ciclista.

2.2. Algoritmo de predicción

Para la predicción de giros de vehículos hacia la derecha, existen varios modelos que se presentan en [21], donde se propone una clasificación de los sistemas de detección en tres niveles:

- Basado en variables físicas [23].
- Basado en maniobras del conductor [7].
- Basado en interacciones e interdependencias de los agentes del tráfico [35].

A continuación se detallan los modelos agrupados por categorías.

2.2.1. Variables físicas

El modelo de conductor inteligente presentado en [22], busca utilizar el *Intelligent Driver Model (IDM)* para predecir los comportamientos de giros vehiculares en intersecciones de calle. Para realizar estas predicciones, utilizan la posición y velocidad del vehículo que se

acerca a la intersección, y siguiendo sus patrones de desaceleración, se ajusta el perfil de desaceleración con el IDM.

El modelo presentado en [36], ofrece predicciones con un éxito de un 90 % en intersecciones generalizadas. Para lograr estos resultados de predicción de giro se basan en *Hidden Markov Model (HMM)*, *Support Vector Machine (SVM)* y *Dynamic Bayesian Network (DBN)*, en conjunto con mapas de las calles de alta precisión. Para obtener estos resultados, obtuvieron datos reales de comportamientos de giros vehiculares en diferentes intersecciones.

La dificultad que presentan estos estudios, es que no consideran los casos en que hay más de un vehículo en la pista y el comportamiento de los conductores cambia ajustándose a los patrones de velocidad y aceleración de los vehículos que los preceden, por lo que es difícil inferir la trayectoria de los vehículos en estos casos.

2.2.2. Maniobras del conductor

El modelo paramétrico presentado en [24], toma en consideración posibles recorridos en vez de maniobras individuales, lo que entrega la posibilidad de combinar diferentes maniobras individuales al momento de predecir el comportamiento del conductor. Además se toma en consideración algunos datos adicionales como el indicador de giro de vehículo y la inclinación de la mirada del conductor para mejorar sus resultados predictivos.

Considerando lo anterior, no podemos en este momento considerar este método, ya que los datos que se requieren no están considerados actualmente dentro del protocolo IEEE 802.11p, lo que dificulta el enviar la información desde el vehículo hasta el punto donde se procesa el algoritmo.

2.2.3. Interacciones e interdependencias de agentes de tráfico

Otra alternativa se presenta en [5], donde logran determinar con cierta certeza, utilizando un modelo "Hidden Markov Models"(HMM) que permite identificar el comportamiento del conductor y las diferentes maniobras que realiza. Para entrenar el modelo utilizan información generada por una "Controller Area Network" (CAN) que entrega información del entorno donde se encuentra el vehículo, por lo que es necesario que esta información esté presente en tiempo real para poder ser accedida y procesada, lo que dificulta el proceso de predicción si es que no se encuentra la información disponible en un ambiente no controlado.

El uso de datos de conducción real para inferir el comportamiento de los conductores [2] ofrece una alternativa interesante ya que utilizando esta información se puede llegar a predecir las maniobras de los conductores y aplicarlo en intersecciones [3], lo que permitiría alertar a ciclistas. Ahora la dificultad que presentan estos modelos, es que todavía no se encuentran bien desarrollados y a la vez, la información que utilizan para realizar las predicciones requiere de información específica que no estará siempre disponible a no ser que todos los vehículos incorporen sensores específicos y los protocolos de comunicación envíen esta información como

estándar.

Es por lo anterior, que para la creación del sistema de predicción se utilizará el modelo propuesto en [23] ya que presenta predicciones en tiempo real y otorga tiempos de predicción de 3 segundos y la información que utiliza es la velocidad, aceleración y posición del vehículo, que es la información que se espera intercambiar como estándar en los protocolos de comunicaciones vehiculares, en conjunto con el modelo de desaceleración para las intersecciones en particular que puede ser calculado previamente.

2.3. Arquitecturas de Comunicación

2.3.1. Redes vehiculares

Dado que el objetivo es transmitir la información fuera del vehículo, es necesario tomar en consideración las redes que posibilitan la comunicación inalámbrica de los vehículos con otros vehículos u otros integrantes de las redes de tráfico.

2.3.2. Sistemas de comunicación

Para los sistemas de comunicación, existen las comunicaciones “vehicle to vehicle” (V2V) [26], “vehicle to infrastructure” (V2I) y “vehicle to everything” (V2X), los cuales utilizan el protocolo de comunicación con el estándar IEEE 802.11p.

En particular las V2V y V2I se pueden utilizar para un espectro amplio de aplicaciones, y se consideran el estándar a utilizar para las aplicaciones de seguridad que se planean implementar en el futuro próximo. Un ejemplo de esto se muestra en [16], donde se utiliza la comunicación V2I para alertar a ciclistas y automovilistas de la presencia de otros usuarios en la vía, utilizando comunicación *Long Term Evolution (LTE)* para informar a ciclistas. Cabe destacar que en este caso, la infraestructura se comunica con los servidores alojados en la nube para informar a los ciclistas por lo que no hay una comunicación V2X directa entre automóviles y ciclistas.

Una alternativa interesante de V2X se presenta en [31] ya que plantea un dispositivo que se coloca en la bicicleta y permite tener comunicación vehicular con IEEE 802.11p directamente, reduciendo los tiempos de comunicación drásticamente en comparación con las otras alternativas, ya que utilizaría los canales dedicados para comunicación vehicular y así el envío de mensajes sería con la máxima frecuencia disponible en la red de comunicación vehicular. En particular, de ser viable esta alternativa, se podría integrar con bluetooth con el dispositivo móvil del ciclista y realizar cualquier procesamiento en dicho dispositivo sin tener que pasar por Internet para realizar la comunicación.

2.4. Aplicaciones de alarma para ciclistas

Una aplicación que ofrece buenos tiempos de comunicación se propone en [16], que ofrece una arquitectura de comunicación pasando por una RSU que se encarga de enviar la información de los vehículos hacia un servidor en la nube para posteriormente notificar a los ciclistas de la presencia de automóviles en la vía. De la misma forma, los ciclistas envían al servidor su información por medio de LTE y este último, envía la información a la RSU para que distribuya los mensajes a los automóviles. Según las validaciones de [15], esta arquitectura aporta a mejorar la seguridad de los ciclistas en las vías y de los vehículos en general, por lo que una arquitectura de solución de este tipo vendría a ser un aporte para resolver la problemática planteada.

Otra alternativa es WifiHonk [9] que utiliza comunicación V2X para un sistema de seguridad para VRU, el cual utiliza Wifi para comunicar directamente a los vehículos con los ciclistas e integra un algoritmo de detección de colisiones que alerta a ciclistas de potenciales accidentes dentro de un marco de tiempo suficiente para que el ciclista alcance a reaccionar. Esta opción es interesante ya que utiliza solamente el teléfono móvil para realizar la comunicación y la predicción, pero se puede mejorar en cuanto al modelo de predicción de colisiones ya que no integra predicción de giros para evitar colisiones en intersecciones sino que considera trayectorias rectas de posibles colisiones.

2.5. Resumen

Dado el estudio del arte realizado y presentado anteriormente, es posible categorizar las investigaciones llevadas a cabo en las áreas de algoritmos de predicción y arquitecturas de comunicación.

Tabla 2.1: Algoritmos de predicción

VARIABLES FÍSICAS	[22], [23], [24]
Maniobras del conductor	[7], [36]
Interacciones e interdependencias de los agentes de tráfico	[2], [3], [5], [35]

Para decidir el algoritmo a utilizar se consideraron los siguientes factores:

1. Los datos de entrada necesarios para el algoritmo deben estar considerados dentro del protocolo de comunicación IEEE 802.11p
2. Debe proveer tiempos de predicción superiores a 2 segundos para ser enviada la alerta por algún canal de comunicación
3. El algoritmo debe considerar un escenario realista, considerando al menos presencia de múltiples vehículos en la calle

Considerando el punto 2, los algoritmos que se basan en interacciones e interdependencias de los agentes de tráfico no califican ya que requieren datos de entrada que no se consideran dentro del protocolo IEEE 802.11p.

Considerando el punto 1, los algoritmos basados en maniobras del conductor, tienen tiempos de predicción inferiores a 2 segundos, por lo que el proceso de enviar las alertas por un canal de comunicación supone una pérdida mayor de tiempo de predicción.

Finalmente, considerando el punto 3, se elige el algoritmo planteado en [23] ya que es el que utiliza posición y velocidad, que están considerados dentro del protocolo IEEE 802.11p, presenta tiempos de predicción sobre 2 segundos y además, considera escenarios con varios vehículos en la vía por lo que supone un escenario mas cercano a la realidad.

Tabla 2.2: Arquitecturas de comunicación

Sistemas de comunicación	[26], [31]
Aplicaciones de alarma para ciclistas	[9], [15], [16]

Para el sistema de comunicación a utilizar, se considera el marco de una ciudad inteligente donde se utilicen sistemas de transporte inteligentes, por lo que se considera que el protocolo IEEE 802.11p es el estándar en este caso. Para el diseño de la arquitectura de comunicación, se utiliza un esquema similar al propuesto en [16].

Capítulo 3

Modelo de Predicción

En este capítulo se presenta el modelo de predicción que se utiliza para detectar y predecir el comportamiento de giro de vehículos hacia la derecha en intersecciones de calle. El modelo seleccionado se traduce a un algoritmo que posteriormente se integra a las arquitecturas de comunicación que se simulan para alertar a los ciclistas.

3.1. Descripción del algoritmo de giro

El algoritmo implementado para este trabajo de tesis, se basa en la descripción provista por [23], donde se detallan los parámetros utilizados y los pasos realizados para calcular la probabilidad de giro de un vehículo. En particular consideramos el método simulado descrito en [23] para determinar el intento de giro y elevar una alerta en caso de un giro hacia la derecha por parte del vehículo. A continuación, se explica en mas detalle la adaptación realizada.

El objetivo del algoritmo es inferir el intento del conductor en intersecciones urbanas. En general, la idea tras este trabajo consiste en crear una trayectoria simulada para cada vehículo, usando modelos y parámetros previamente definidos. La trayectoria simulada es comparada con la trayectoria real del vehículo desde unos segundos antes de llegar a la intersección, para estimar la probabilidad de cada hipótesis posible de trayectoria predefinida. La distribución de probabilidad obtenida es usada para inferir la intención del conductor y la trayectoria futura del vehículo. Las posibles intenciones del conductor se definen como sigue:

- I_1 : Seguir recto
- I_2 : Detenerse en luz roja
- I_3 : Girar a la derecha
- I_4 : Girar a la derecha y detenerse en el cruce peatonal

Para cada intento uno esperaría diferentes conjuntos de perfiles de velocidad deseada y para cada tipo de perfil de velocidad deseada una distribución de probabilidad diferente para el parámetro de aceleración máxima a . La combinación de perfiles y aceleración máxima se presentan en la Tabla 3.1.

Tomando esto en consideración, el primer paso es modelar la aceleración del vehículo \dot{v} usando *Intelligent Driver Model (IDM)* [23] :

$$\dot{v} = a \left[1 - \left(\frac{v}{u} \right)^\delta - \left(\frac{d^*(v, \Delta v)}{d} \right)^2 \right] \quad (3.1)$$

$$d^*(v, \Delta v) = d_0 + Tv + \frac{v\Delta v}{2\sqrt{ab}} \quad (3.2)$$

Tabla 3.1: Hipótesis de inferencia de intención del conductor

Intención I	H: Modelo X $a(m/s^2)$
I_1 : Seguir recto	{1, 2, 3} X {1.5, 2.0, 2.5}
I_2 : Detenerse	{1, 2, 3} X {1.5, 2.0, 2.5}
I_3 : Girar a la derecha	{1, 2, 3} X {1.5, 2.0, 2.5}
I_4 : Girar a la derecha pero detenerse	{1, 2, 3} X {1.5, 2.0, 2.5}

Los valores correspondientes a cada parámetro se dan en la Tabla 3.2. Cuando no hay un vehículo que precede presente, la aceleración calculada \dot{v} es determinada solamente por el parámetro de aceleración máxima a , la velocidad v , la velocidad deseada u , y un exponente de aceleración fijado δ .

Tabla 3.2: Parámetros del Modelo de Conductor Inteligente de acuerdo a [23]

Parameter	Value
Aceleración máxima a	0..5 m/s^2
Exponente de aceleración δ	4
Velocidad deseada u	0..60 km/h
Desaceleración cómoda b	3 m/s^2
Brecha mínima con el vehículo que precede d_0	2.0 m
Brecha de tiempo con el vehículo que precede T	0.8 s

Parámetros tales como distancia mínima d_0 o intervalo de tiempo T provienen calculadas desde variables de tránsito. En [23], el *IDM* se usa junto a tres perfiles de conducción diferentes, abarcando desde estilo defensivo hasta estilo de conducción deportista. Para cada estilo de conducción, el modelo considera tres aceleraciones máximas diferentes a : 1.5 m/s^2 para defensivo, 2.0 m/s^2 para regular, y 2.5 m/s^2 para estilo deportista de conducción. La distribución de probabilidad en [23] es calculada para cada modelo y parámetro de aceleración a . Los valores resultantes de estos cálculos se usan posteriormente para inferir la intención del conductor.

En esta adaptación y con la finalidad de reducir el tiempo de procesamiento, se propone simplificar las opciones por medio de la reducción de intenciones posibles del conductor dejando solo I_1 y I_3 , las cuales son: seguir derecho y girar hacia la derecha, respectivamente. A continuación, se procede a calcular la distribución de probabilidad que se puede modelar como una red Bayesiana, basada en el intento actual del conductor I , la hipótesis aplicable H , y una observación O que todavía no está definida:

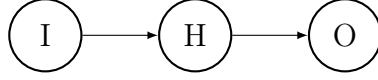


Figura 3.1: Bayesian network

La probabilidad para una intención particular I_j dada la observación O , se puede escribir como:

$$P(I_j|O) = \sum_i P(I_j|H_i)P(H_i|O), \quad (3.3)$$

donde $P(I_j|H_i)$ es 1 o 0 dependiendo de H_i . Las probabilidades para las hipótesis individuales se calculan como sigue:

$$P(H_i|O) = \frac{P(O|H_i)P(H_i)}{\sum_j P(O|H_j)P(H_j)} \quad (3.4)$$

Las probabilidades anteriores $P(H_i)$ se pueden obtener desde:

$$P(H_i) = P(I_j)P(M_k)P(a_l|M_k), \quad (3.5)$$

donde I_j es el intento, M_k es el modelo de velocidad deseada, y a_l es el parámetro de aceleración longitudinal máxima asociada con la hipótesis H_i . De esta forma, $P(M_k)$ y $P(a_l|M_k)$ se puede obtener por medio del gráfico de distribución de probabilidad que se presenta en la Figura 3.2. La distribución del intento U es considerado uniforme por lo que $P(I_j) = 0,5 \forall j$.

Ahora, el enfoque de simulación propuesto en [23] es seleccionado. Considerando \hat{s} y \hat{v} como la posición y velocidad simulada de un vehículo, se pueden comparar estos valores con los valores reales observados de conductores, para obtener la distribución de probabilidad. La función de densidad de probabilidad f_{AS} para la hipótesis H_i se define en [23] como:

$$f_{AS}(a(t), s(t)|H_i) = \frac{1}{2\pi\sigma_s\sigma_v} \exp\left(-\frac{1}{2}e^2\right), \quad (3.6)$$

donde

$$e = \sqrt{\left(\frac{s(t) - \hat{s}(t)}{\sigma_s}\right)^2 + \left(\frac{v(t) - \hat{v}(t)}{\sigma_v}\right)^2}. \quad (3.7)$$

Usando el teorema de Bayes para las funciones de densidad de probabilidad, la probabilidad $P(H_i|a(t), s(t))$ se determina entonces como:

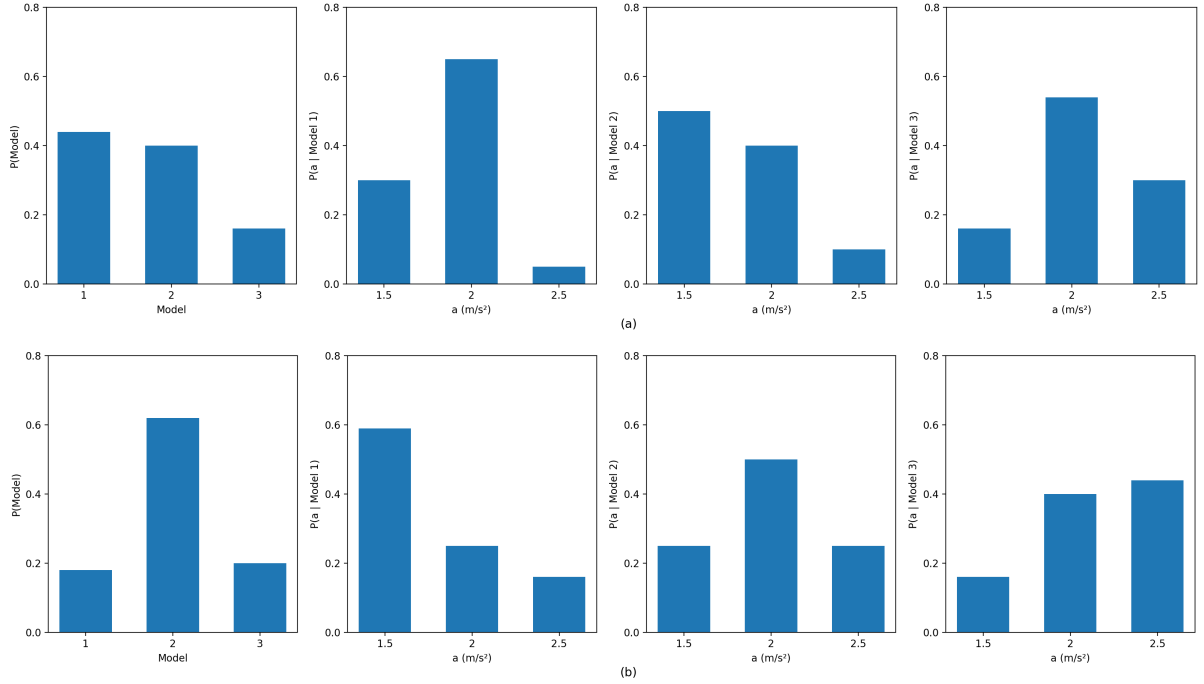


Figura 3.2: Distribución de probabilidad para cada modelo M y parámetro de aceleración a dado que la parametrización fue obtenida por *clustering* (a) o desde el trazado de curvatura (b). Figura basada en [23]

$$P(H_i|a(t), s(t)) = \frac{f_{AS}(a(t), s(t)|H_i)P(H_i)}{\sum_j f_{AS}(a(t), s(t)|H_j)O(H_j)} \quad (3.8)$$

Los autores definen $\sigma_s = 1,2m$ y $\sigma_v = 1,2m/s$ como valores obtenidos a partir de evidencia empírica [23]. Usar valores apropiados para σ_s y σ_v es importante ya que tienen mayor influencia en cuan fácil se favorece una hipótesis sobre las otras.

3.2. Adaptación para procesamiento en línea

El algoritmo estudiado se enfoca en medir el comportamiento de un conductor de un vehículo y alertarlo en caso de que vaya a realizar un giro a la derecha para prevenir un potencial accidente con algún usuario vulnerable de las vías. En esta tesis se busca utilizar el mismo algoritmo pero cambiando el foco de enviar la alerta a un ciclista como un método de prevención activa de accidentes.

Considerando el contexto de una ciudad inteligente con sistemas inteligentes de transporte, los vehículos utilizan el estándar IEEE 802.11p para sus comunicaciones vehiculares. Tomando este contexto, se puede extraer el procesamiento del algoritmo del vehículo y trasladarlo a una RSU, un servidor en la nube o a un dispositivo integrado en la bicicleta del ciclista. Con sacar el procesamiento del vehículo se evita el tener que implementar en cada vehículo el

algoritmo y solo se necesita que los vehículos cuenten con el estándar IEEE 802.11p y envíen la información estándar considerada dentro del protocolo.

Por lo anterior, se adapta el algoritmo para que reciba los datos de entrada por medio de una red. El algoritmo se adaptó para recibir mensajes con frecuencia de 10 Hz y tolerar la pérdida de paquetes como se puede comprobar mas adelante al analizar las arquitecturas propuestas.

3.2.1. Complejidad espacial

Para el procesamiento en línea se debe considerar que el algoritmo recibe los datos de entrada provenientes de una red. Dado esto, se debe guardar el estado anterior calculado por el algoritmo y los resultados de la estimación del siguiente paso para poder mantener la heurística del algoritmo.

Los datos del estado anterior son los siguientes:

- Posición anterior
- Posición siguiente estimada
- Velocidad anterior
- Velocidad siguiente estimada
- Probabilidad de giro

Con estos datos almacenados y los que va recibiendo el algoritmo se calcula cada paso de la predicción por lo tanto, el espacio utilizado por el algoritmo es de orden constante $\mathcal{O}(1)$.

3.2.2. Complejidad temporal

Para analizar la complejidad temporal del algoritmo en su forma fuera de línea, se podría considerar el peor caso y por consecuencia el tiempo sería de orden lineal $\mathcal{O}(n)$. Por otra parte, analizar el peor caso para el algoritmo en línea, considerando que los datos de entrada llegan por una red de comunicación, no sería beneficioso ya que en el peor caso los datos podrían no llegar.

Es por lo anterior que se procede a realizar un análisis amortizado del algoritmo para determinar la complejidad temporal del algoritmo.

Considerando la descripción anterior del algoritmo, se observa que el tiempo de evaluar cada paso del algoritmo es de tiempo constante $\mathcal{O}(1)$. Por otra parte, cada mensaje con datos que recibe el algoritmo trae un retardo adicional que va a depender de los canales de comunicación que se utilicen para enviar cada mensaje. La variabilidad del retardo de cada mensaje puede ser alta, pero de todas maneras se puede obtener el promedio y distribuirlo para cada mensaje dejando un retardo de \mathcal{D} para cada mensaje. Con todo lo anterior, tenemos que para procesar todos los mensajes hasta entregar una alerta, se requiere de tiempo $\mathcal{O}(n)$

y además, el tiempo que se demoran en llegar los mensajes es $\mathcal{O}(Dn)$.

Calculando el tiempo amortizado tenemos:

$$\mathcal{O}(n) + \mathcal{O}(Dn) \tag{3.9}$$

$$\mathcal{O}(n + Dn) \tag{3.10}$$

$$\mathcal{O}((D + 1)n) \tag{3.11}$$

$$\mathcal{O}(n) \tag{3.12}$$

Finalmente, tenemos que el tiempo amortizado es $\mathcal{O}(n)$.

3.3. Adaptaciones de implementación del algoritmo

El proceso de cálculo de probabilidades para determinar giros presentado en [23], se interpreta y adapta para poder incluirlo en el simulador SUMO y realizar los estudios pertinentes utilizando el Framework Veins.

Los paquetes de datos viajan con frecuencia de 10 Hz desde el vehículo utilizando el protocolo IEEE 802.11p y llevan la información de posición y velocidad del vehículo en ese instante. Estos paquetes se formatean como "DemoSafetyMessage" del protocolo IEEE 802.11p para ser procesados en el nodo correspondiente, ya sea la RSU o el nodo en la bicicleta.

El primer paquete para procesar se almacena y desde el segundo paquete se comienza a procesar, ya que se realizan comparaciones entre los datos del paquete actual y del anterior para realizar los cálculos. Cuando el nodo de procesamiento tiene los datos necesarios, comienza a calcular las probabilidades de giro para cada instante de tiempo, asumiendo una frecuencia constante de entrega de los mensajes. El beneficio de asumir una frecuencia constante es que, simplifica los cálculos necesarios para poder predecir el giro y el impacto negativo es despreciable para la predicción de giros.

Para el cálculo de la predicción de giros, se toman cinco hipótesis de las nueve calculadas según la matriz de los modelos con las diferentes aceleraciones. Al tomar estas cinco hipótesis ponderadas con su probabilidad y la probabilidad de la densidad, se procede a dividir cada probabilidad de hipótesis por el valor total de las probabilidades para posteriormente sumar los resultados, dando el valor de la probabilidad de giro.

Luego de varias pruebas, se determina que el valor de la probabilidad de giro debe ser superior a 0,689 para optimizar la predicción, disminuyendo los falsos positivos y los falsos negativos. Una vez se detecta un posible giro, se envía la alerta correspondiente al nodo del ciclista para que el ciclista frene. En caso de que el procesamiento se realice en el nodo del ciclista, dicho nodo levanta la alerta para que el ciclista frene. Este proceso se realiza para cada paquete que se recibe en el nodo de procesamiento, por lo que el algoritmo se va ejecuta en tiempo real.

Capítulo 4

Arquitectura de Comunicación

Con el fin de alertar a los ciclistas a tiempo para prevenir accidentes, en este capítulo se estudia y comparan diferentes arquitecturas de comunicación que permitan alertar a los ciclistas y de esta forma, determinar los escenarios y arquitecturas que en conjunto con el algoritmo de predicción de giro, permitirían disminuir los accidentes con ciclistas involucrados.

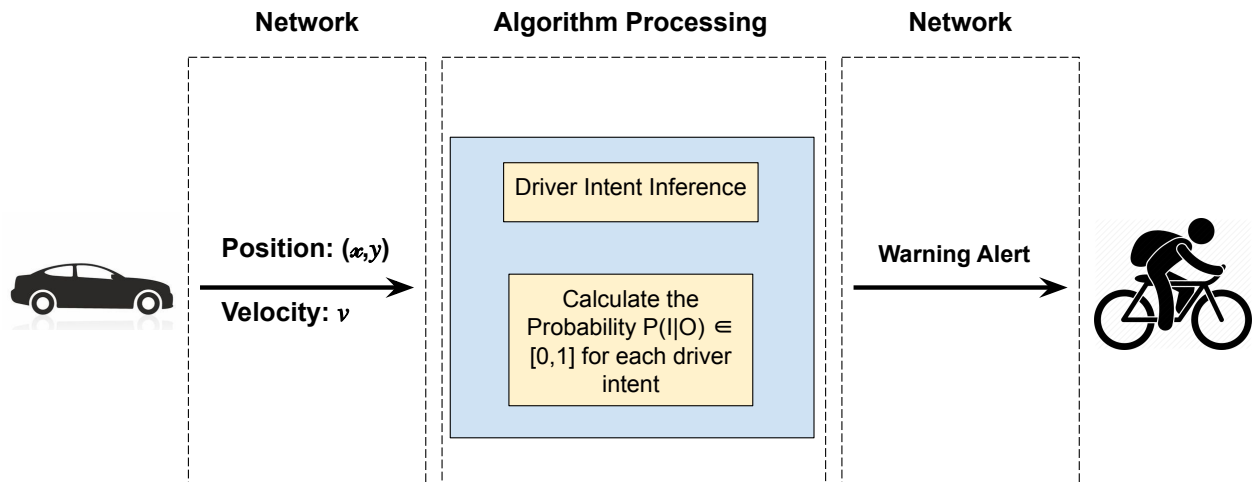


Figura 4.1: Arquitectura general propuesta de comunicación vehículo-ciclista

El escenario general de arquitectura de comunicación que se plantea, se presenta en la figura 4.1 donde se muestra que la información es enviada desde el vehículo, pasa por una red de comunicación y llega al ente que procesa el algoritmo de predicción de giro y posteriormente, se envía por otra red de comunicación la alerta al ciclista sobre la potencial colisión. Cabe destacar que para las arquitecturas que se estudian, podría ser que solo se requiera una red y el algoritmo de predicción sea ejecutado directamente en el ciclista.

Para definir las arquitecturas de comunicación a estudiar, se presentan primeramente los estándares de comunicación que se utilizarán para la construcción de las diferentes arquitecturas. Para esto, se presentan las redes vehiculares ad hoc (VANET) para posteriormente describir el estándar de acceso inalámbrico en ambientes vehiculares (WAVE), el cual corresponde a un conjunto de protocolos que se definieron en la IEEE para las comunicaciones

vehiculares inalámbricas. Además, se presentará una descripción del estándar LTE que será utilizado en algunas arquitecturas para llevar los mensajes al ciclista.

4.1. VANET

Una red *ad hoc* vehicular [39], habitualmente referida por su acrónimo en inglés VANET, es un tipo de red de comunicación que utiliza a los vehículos como nodos de la red. Dado el reducido alcance del canal de comunicación (hasta 300 metros), la conectividad se establece de forma esporádica (*ad hoc*). Por este motivo, estas redes se consideran un tipo específico de red móvil de comunicación llamado Mobile Ad hoc Network (MANET). No obstante, este tipo de redes tienen una serie de retos específicos asociados, tales como la alta volatilidad de las redes, la velocidad de los nodos comunicantes y la concentración de los nodos en un área.

El caso de estudio al que nos enfocaremos, es al estándar de *Wireless Access in Vehicular Environments (WAVE)* que es un conjunto de protocolos de comunicación inalámbricas propuestos por la IEEE, ya que dicho protocolo ha tomado fuerza a nivel internacional en lugares como Estados Unidos, Europa y Japón ya que predominan la industria automovilística y se busca que sea el estándar internacional para las comunicaciones vehiculares inalámbricas.

4.1.1. WAVE

Wireless Access in Vehicular Environments (WAVE) [20], es la estandarización de un grupo de protocolos de acceso inalámbrico en entornos vehiculares llevada a cabo por un grupo de trabajo del IEEE. El objetivo principal de WAVE, es proporcionar comunicación ya sea V2V (vehículo a vehículo), como V2I (vehículo a infraestructura) en un entorno rápidamente cambiante, en el cual, el intercambio de información debe ser completada en un periodo de tiempo muy corto. Algunas de las aplicaciones son la gestión de autovías, prevención de accidentes, gestión del tiempo de ruta, gestión de emergencias, información al viajero, etc.

Este estándar, toma principalmente los protocolos IEEE 802.11p para la gestión de las capa Física y la capa de Control de Acceso al Medio o MAC por su acrónimo de *Media Access Control*. Además, para las otras capas, utiliza el estándar IEEE 1609 con sus normas para las capas respectivas como se presenta en la figura 4.2.

4.1.2. IEEE 802.11p

El estándar IEEE 802.11p [11] (derivado del estándar IEEE 802.11), describe el uso de las capas inferiores (la capa física y la capa básica MAC) de la pila WAVE. Las capas superiores, están definidas por el estándar IEEE 1609.

Como se presenta en [41], los parámetros de IEEE 802.11p son los que se muestran en la tabla 4.1.

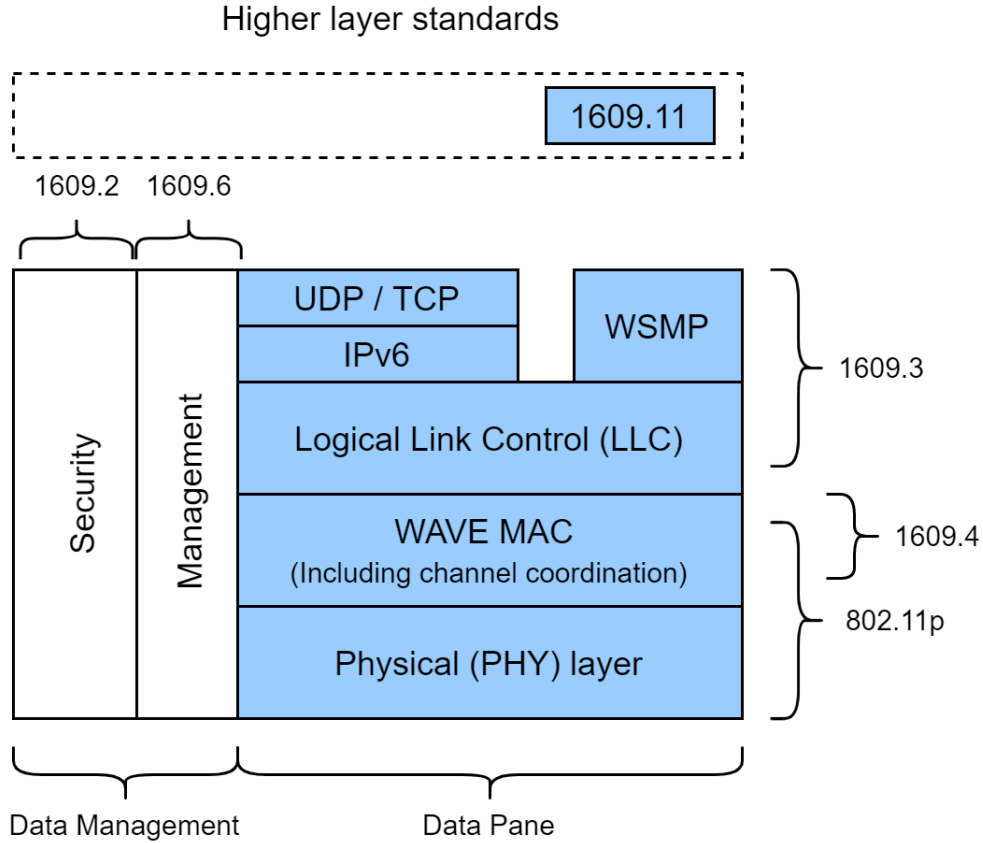


Figura 4.2: Pila WAVE

Tabla 4.1: Parámetros de IEEE 802.11P. Basada en [41]

Parameters	IEEE 802.11p	Notations
Tslot	13 μ s	Un espacio de tiempo
Lodging	2 μ s	Retraso de propagación
TP	32 μ s	Tiempo de transmisión del preámbulo físico
TDIFS	58 μ s	Tiempo DIFS
TSIFS	32 μ s	Tiempo SIFS
CWmin	15	Tamaño mínimo de ventana de retroceso
TPHY	64 μ s	Tiempo de transmisión del encabezado PHY
TSYM	8 μ s	Tiempo de transmisión para un símbolo
LH_DATA	28bytes	Sobre costo de MAC en bytes
LACK	14bytes	Tamaño de ACK en bytes
TH_DATA		Tiempo de transmisión sobre costo de MAC
TACK		Tiempo de transmisión de ACK
LDATA		Carga útil en bytes
TDATA		Tiempo de transmisión para la carga útil
Mbps	3, 4.5, 6, 9, 12, 18, 24, 27	Millón de bits por segundo
MDBPS	24,36,48,72,96,144,192,216	Bits de datos por OFMD

4.1.3. IEEE 1609

El estándar IEEE 1609 está compuesto por un conjunto de normas las cuales definen la arquitectura, el modelo de comunicación, la estructura de gestión, mecanismos de seguridad y el acceso físico para las comunicaciones en el entorno WAVE.

Estándar IEEE 1609.2

Este estándar especifica la seguridad de las comunicaciones para el entorno WAVE para las aplicaciones y procesos que se ejecutan en las distintas capas de la pila de protocolos. Estos servicios de seguridad consisten en:

- Servicios de procesamiento seguros: proporcionan un procesamiento seguro para permitir comunicaciones seguras de datos así como de los *WAVE Service Advertisement (WSA)*.
- Servicios de gestión seguros: proporciona servicios de gestión de certificados y gestiona la validación de los mismos. También gestiona la información relativa a las claves privadas usadas en el envío de WSAs seguros, así conseguiremos WSAs firmados los cuales podrán ser verificados en la recepción de los mismos.

Para una comunicación segura de los datos, en la transmisión hay que generar los datos firmados y cifrados, y en la recepción se verificará la firma y se descifrarán los datos. Para las operaciones de criptografía, los métodos usados serán: claves privadas y claves públicas asociadas, claves privadas y certificados asociados, y el uso de certificados digitales. Los mecanismos criptográficos que soporta este estándar son: ECDSA (Elliptic Curve Digital Signature Algorithm) y ECIES (Elliptic Curve Integrated Encryption Scheme).

Estándar IEEE 1609.3

Este estándar define las operaciones de servicios de red y de transporte, capa 3 (capa de red) y capa 4 (capa de transporte) de la pila de comunicaciones *Open Systems Interconnection (OSI)*. Estos servicios de red, están diseñados para soportar comunicaciones inalámbricas entre los dispositivos WAVE. Este estándar se compone de las capas del plano de datos y la entidad del plano de gestión asociado *WAVE Management Entity (WME)*.

Los componentes del plano de datos especificados por esta norma son: *Logical Link Control (LLC)*, IPv6 y las capas superiores como *Transmission Control Protocol (TCP)* y *User Datagram Protocol (UDP)*, y por último *WAVE Short Message Protocol (WSMP)*. Respecto al plano de gestión, la entidad de gestión WAVE (WME), proporciona las siguientes funcionalidades:

- Solicitud de servicio y asignación de acceso al canal.
- Entrega de datos de gestión.
- Monitorización y verificación de WSAs.

- Configuración de IPv6.
- Mantenimiento de la *Management Information Base (MIB)* la cual contiene información de configuración y estado.

Estándar IEEE 1609.4

Este estándar se encarga de la coordinación de canales y se compone de características adicionales para operaciones OCBEnabled en la subcapa de MAC especificada en IEEE 802.11p. OCBEnabled indica el funcionamiento fuera del servicio básico especificado por IEEE 802.11p. Tanto el plano de datos como el de gestión, están especificados en este estándar. Además de utilizar los servicios prestados por la norma para la coordinación de los canales, también sirve de apoyo para la entrega de MSDU (MAC Service Data Unit). La capa MAC y la capa física, son gestionadas por diferentes entidades de gestión, las cuales son:

- *MAC Layer Management Entity (MLME)* para la capa MAC.
- *PHY Layer Management Entity (PLME)* para la capa física.

Respecto al servicio de plano de datos se especifica lo siguiente:

- La coordinación de canal: la subcapa MAC coordina los intervalos de canal para que los paquetes sean transmitidos en el canal adecuado y en el momento adecuado.
- El enrutamiento en el canal: esta especificación incluye el enrutamiento de paquetes de datos de la LLC en el canal designado, y el establecimiento de parámetros para las comunicaciones WAVE.
- La prioridad de usuario: es importante que se establezca distintos niveles de prioridad, ya que WAVE es usado por aplicaciones de seguridad y de no seguridad (por ejemplo, aplicaciones para el entretenimiento). Se establecen hasta ocho niveles de prioridad asignada en los parámetros EDCA (Enhanced Distributed Channel Access). Con EDCA, el tráfico de alta prioridad tiene mayor posibilidad de ser enviado que el tráfico de baja prioridad.

Respecto a los servicios del plano de gestión se especifica:

- La sincronización multicanal: el objetivo de este, es alinear los intervalos del canal entre las comunicaciones de los dispositivos WAVE.
- El acceso a canales: el MLME es el encargado de controlar el acceso a los canales radio. El MLME será el encargado de recibir y generar tramas VSA (Vendor Specific Action). Mantenimiento de una MIB la cual contiene información de configuración y estado.
- Otros servicios de IEEE 802.11: el MLME permite el acceso a servicios de este estándar los cuales pueden ser invocados por un canal.

4.1.4. LTE

Long-Term Evolution (LTE)[1] , es un estándar para la comunicación inalámbrica de banda ancha para dispositivos móviles y terminales de datos, basado en las tecnologías GSM / EDGE y UMTS / HSPA. Aumenta la capacidad y la velocidad utilizando una interfaz de radio diferente junto con mejoras en la red central. LTE es la ruta de actualización para operadores con redes GSM / UMTS y redes CDMA2000. Las diferentes frecuencias y bandas LTE utilizadas en diferentes países significan que solo los teléfonos multibanda pueden usar LTE en todos los países donde se admite.

4.1.5. Métricas de evaluación

Time To Collision - TTC

Como se describe en [17] y [30], *Time To Collision (TTC)* es un indicador basado en tiempo para detectar potenciales accidentes en las evaluaciones de la seguridad del tráfico.

Para este caso, utilizar el TTC sirve para determinar el tiempo con que la arquitectura de alerta informa al ciclista de un potencial accidente. Para saber si este TTC entregado por la arquitectura de solución es útil, se debe considerar un tiempo mínimo de reacción del ciclista para que pueda realizar una maniobra evasiva y de esta forma mitigar o prevenir un potencial accidente.

Como el objetivo de la arquitectura de alerta temprana es prevenir la mayor cantidad de accidentes posibles, se busca colocar requerimientos estrictos en cuanto al tiempo que necesita un ciclista para reaccionar y maniobrar para prevenir un accidente. Dando lo anterior, se utilizan los peores casos de los rangos presentados en las publicaciones estudiadas.

Según se presenta en [6], dentro de los diferentes escenarios de estudio, el tiempo de reacción de los ciclistas varía en torno a los 300 milisegundos y 650 milisegundos por lo que para nuestro caso, se considerarán 650 milisegundos como el tiempo de reacción de los ciclistas.

Según diferentes publicaciones [4], [12], [32], [18] la velocidad de los ciclistas que se desplazan por ciclovías varía entre 13 km/h y 30 km/h. Según lo presentado en la Tabla 4.2, la tasa de desaceleración en pavimento seco y sin pendiente es de 4.8 m/s^2 .

Como se puede observar en la Tabla 4.3, el tiempo requerido va desde 1.40 segundos a 2.38 segundos, para que los ciclistas puedan maniobrar y frenar por completo. Esto es considerando un escenario donde el pavimento está seco.

Tabla 4.2: Criterios clave de rendimiento. Basada en [28]

Bicyclist Type	Feature	Value	
		U.S. Customary	Metric
Typical upright adult bicyclist	Speed, paved level terrain	8-15 mph	13-24 km/h
	Speed, downhill	20-30 plus mph	32-50 plus km/h
	Speed, uphill	5-12 mph	8-19 km/h
	Perception reaction time	1.0-2.5s	1.0-2.5s
	Acceleration rate	1.5-5.0 ft/s^2	0.5-1.5 m/s^2
	Coefficient of friction for braking, dry level pavement	0.32	0.32
	Deceleration rate (dry level pavement)	0.16 ft/s^2	4.8 m/s^2
Recumbent bicyclist	Deceleration rate for wet conditions (50-80 % reduction in efficiency)	8.0-10.0 ft/s^2	2.4-3.0 m/s^2
	Speed, level terrain	11-18 mph	18-29 km/h
	Acceleration rate	3.0-6.0 ft/s^2	1.0-1.8 m/s^2
	Deceleration rate	10.0-13.0 ft/s^2	3.0-4.0 m/s^2

Tabla 4.3: Tiempo total requerido por un ciclista para frenar completamente según la velocidad, considerando tasa de desaceleración igual a $4.8 m/s^2$

Velocidad km/h	Tiempo Frenado	Tiempo Total Requerido
13 km/h	0.75s	1.40s
17.28 km/h	1s	1.65s
20 km/h	1.15s	1.8s
25 km/h	1.44s	2.09s
30 km/h	1.73s	2.38s

Precisión de la predicción

Otro factor importante a considerar, es la precisión con que el algoritmo de giro logra predecir los giros a la derecha. Para esto, se define la precisión como:

$$\text{Precisión} = \frac{\text{VerdaderosPositivos} + \text{VerdaderosNegativos}}{\text{MuestrasTotales}}, \quad (4.1)$$

Donde

- VerdaderosPositivos: Maniobras correctamente detectadas como comportamiento de giro hacia la derecha
- VerdaderosNegativos: Maniobras correctamente detectadas como comportamiento de continuar recto por la ruta
- MuestrasTotales: La cantidad total de rutas analizadas

Para obtener la precisión, construiremos la matriz de confusión para el algoritmo de predicción de giro con los resultados obtenidos de analizar las rutas de los vehículos que se estudiarán, utilizando los parámetros ya definidos anteriormente. La matriz de confusión quedará como se presenta en la Tabla 4.4

Tabla 4.4: Matriz de confusión para giros hacia la derecha

n = Cantidad rutas	Real: Girar Derecha	Real: Seguir Recto
Predicho: Girar Derecha	Verdaderos Positivos	Falsos Positivos
Predicho: Seguir Recto	Falsos Negativos	Verdaderos Negativos

Además de la precisión de la predicción, se considera el impacto que tienen los falsos positivos y los falsos negativos, ya que:

- Falso positivo: Predice un giro hacia la derecha de forma incorrecta y el vehículo no gira y sigue recto
- Falso negativo: Predice que el vehículo no gira hacia la derecha y seguirá recto, pero el vehículo si realiza un giro hacia la derecha

Como los falsos positivos predicen un giro que no es real, un ciclista que recibe la alerta realizaría una maniobra evasiva para evitar un accidente que no va a ocurrir. Si bien considerando que la arquitectura de solución busca evitar accidentes y una alerta de un accidente que no va a ocurrir no es mas que un contrat tiempo para el ciclista, se busca disminuir la cantidad de falsos positivos para que el ciclista no pierda la confianza en la arquitectura de solución.

Por otra parte, los falsos negativos son el caso mas perjudicial para los ciclistas, ya que, no alertan al ciclista cuando el vehículo va a realizar un giro hacia la derecha, por lo que deja

al ciclista completamente expuesto a un posible accidente. En este caso es muy importante disminuir la mayor cantidad de falsos negativos posibles, ya que, por cada falso negativo, podría producirse una colisión entre el ciclista y un vehículo.

Capítulo 5

Evaluación de arquitecturas de comunicación

5.1. Origen de los Datos

En esta sección se presentan los datos utilizados para simular las rutas vehiculares con las que se probarán las arquitecturas de comunicación y el algoritmo de predicción integrado en las simulaciones. Se presentará el origen y el formato de donde se obtienen los datos y posteriormente se detalla el procesamiento de los datos que se realiza para poder entregarlos como entrada para el algoritmo de predicción de giros.

5.1.1. TAPAS Cologne Scenario

El escenario de simulación “TAPAS Cologne” describe el tráfico de un día completo en la ciudad de Colonia, Alemania. La solicitud original de los datos proviene de TAPAS¹, un sistema que calcula los deseos de movilidad para un área de población. Estos cálculos son realizados en base a la información sobre los hábitos de viaje de los alemanes y en la infraestructura de la zona en que viven. Los datos originales usaban una red de carreteras de código cerrado y se asignaron a una rama del proyecto OpenStreetMap.

El paquete del escenario TAPAS-Cologne contiene:

- Red de carreteras importadas desde OSM²
- Puntos de interés y polígonos importados desde OSM
- Viajes mapeados entre 6:00 am y 8:00 am
- Viajes mapeados entre 00:00 h y 23:59 h
- El archivo SUMO-configuration

¹Travel and Activity Patterns Simulation

²OpenStreetMap

Para esta tesis, se utilizan los viajes mapeados entre 6:00 am y 8:00 am que consiste en 252731 rutas. Estas rutas se encuentran en el archivo `cologne6to8trips.xml` [37].

5.1.2. Formato de los Datos

Las rutas están contenidas en archivos con formato *Extensible Markup Language (XML)*. Cada ruta está definida dentro de un tag `<trip/ >` con sus parámetros definidos como atributos. El contenido de cada ruta se describe en la Tabla 5.1.2.

Atributo	Descripción	Ejemplo
id	Identificador de la ruta	"34758_353_0"
depart	Tiempo en que el vehículo ingresa a la red	"21601.00"
from	Calle de inicio	23074654#0"
to	Calle de destino	232236528#1"
type	Tipo de vehículo	"pkw"

Tabla 5.1: Descripción de atributos de un viaje

Esta información es suficiente para poder ejecutar la simulación en SUMO y que los vehículos se desplacen a lo largo de la red. Pero esta información no es suficiente para que el algoritmo de predicción de giros pueda funcionar, por lo que es necesario procesar y trabajar los datos para obtener los datos de entrada necesarios. En este caso, para cada ruta, se busca obtener información de posición y velocidad en cada instante de tiempo con una frecuencia de 10 Hz.

5.2. Procesamiento de los Datos

A continuación se describe el proceso necesario para obtener los datos finales que nos permitirán probar nuestra integración de algoritmo de predicción en conjunto con las diferentes arquitecturas propuestas para formar la arquitectura de prevención activa para los ciclistas.

El primer paso, es obtener las rutas para cada vehículo. Con estas rutas, podremos seleccionar los vehículos que pasan por la intersección deseada. Para obtener las rutas, se ejecuta la simulación con los parámetros presentados en la Tabla 5.2.

Parámetro	Descripción
<code>-vehroute-output <FILE></code>	Genera un archivo de nombre <code><FILE></code> que contiene las rutas de cada vehículo
<code>-step-length 0.1</code>	La simulación calcula el estado de los vehículos cada 0.1 segundos

Tabla 5.2: Parámetros de ejecución y sus descripciones

La simulación genera como salida un archivo que contiene todas las rutas agrupadas por

vehículos. Cada ruta se define dentro del tag <vehicle>y sus atributos se presentan en la Tabla 5.3.

Atributo	Descripción	Ejemplo
id	Identificador del vehículo	"39732_360_0"
type	Tipo del vehículo	"pwk"
depart	Instante de tiempo en que el vehículo ingresa a la red	"21601.00"
arrival	Instante de tiempo en que el vehículo llega a destino	"21608.00"
replacedOnEdge	Arista en la que se reemplazó la ruta	266542217"
replacedAtTime	Instante de tiempo en que se realizó el reemplazo	"21635.00"
edges	Aristas por las cuales el vehículo pasó	35954406#4 -35954406#2"

Tabla 5.3: Atributos asociados a rutas con aristas incluidas

Ejecutar la simulación con todas las rutas, implica que SUMO debe estar calculando la información para cada vehículo presente en la red para cada instante de tiempo. El intervalo de tiempo es de 100 milisegundos y por cada intervalo se realiza el calculo completo de la red y los vehículos en ella. Considerando que esta red es de una ciudad completa y la cantidad de rutas es elevada, el tiempo de procesamiento va creciendo por cada intervalo de tiempo. Para acelerar el proceso de simulación, se segmentan las rutas en grupos de diez mil rutas y se ejecuta la simulación con cada bloque de diez mil rutas. Con este ajuste el tiempo de ejecución se reduce considerablemente ya que la cantidad de vehículos simultáneos en la red disminuye y baja la carga de procesamiento para cada intervalo de tiempo del procesador. Realizando este procedimiento se acorta el tiempo de procesamiento total y se obtienen las rutas con la trayectoria por la cual pasan los vehículos, incluyendo los identificadores de las calles. Con esta información se puede filtrar las rutas según las calles por las que han pasado.

Tomando las rutas, se procede eliminar las rutas reemplazadas dejando solamente la ruta final que tomó cada vehículo. Para esto, se ejecuta el Script : remove_unused_routes.py y el resultado obtenido se presenta en la Tabla 5.4.

Atributo	Descripción	Ejemplo
id	Identificador del vehículo	"39732_360_0"
type	Tipo del vehículo	"pwk"
depart	Instante de tiempo en que el vehículo ingresa a la red	"21601.00"
arrival	Instante de tiempo en que el vehículo llega a destino	"21608.00"
edges	Aristas por las cuales el vehículo pasó	35954406#4 -35954406#2"

Tabla 5.4: Atributos asociados a rutas con aristas finales

Al obtener una ruta por cada vehículo, se identifican potenciales intersecciones que cuenten con suficientes rutas. El proceso de detección es en primera instancia, observar la topología de la red dentro del simulador, detectar las intersecciones que cumplen con los requisitos necesarios para que se presente el comportamiento deseado (giros hacia la derecha y seguir recto), y posteriormente se obtiene el identificador de la calle por la que los vehículos ingresan a la intersección. Con este id se realiza una búsqueda dentro del archivo que contiene todas las rutas y se contabiliza la cantidad de vehículos que pasan por dicha intersección.

Una vez detectada una intersección deseada, se procede a extraer todas las rutas que pasan por dicha intersección utilizando el script `get_specific_routes.py`

Con esto se obtienen solo las rutas necesarias dentro de la red para analizar el comportamiento de los vehículos cuando pasan por esa intersección específica. Una vez obtenidas las rutas deseadas, se procede a ejecutar la simulación con los parámetros `-fcd-output` y `-step-length 0.1`. La descripción y funcionalidad de los parámetros se presenta en la Tabla 5.5.

Parámetro	Descripción
<code>-fcd-output <FILE></code>	Genera un archivo de nombre <code><FILE></code> que contiene las trazas GPS de las rutas
<code>-step-length 0.1</code>	La simulación calcula el estado de los vehículos cada 0.1 segundos

Tabla 5.5: Parámetros de ejecución y sus descripciones

El contenido del archivo `FILE.xml` contiene la información agrupada por `<timestep/>`y para cada instante de tiempo agrupa la información de cada vehículo en etiquetas `<vehicle/>`cuyos parámetros se detallan en la Tabla 5.6

Parámetro	Descripción	Ejemplo
<code>id</code>	Identificador del vehículo	"39732_360_0"
<code>x</code>	Tipo del vehículo	"11859.52"
<code>y</code>	La coordenada Y absoluta del vehículo medida en el centro del parachoques delantero. EL valor depende de la proyección geográfica dada.	"12492.48"
<code>angle</code>	El ángulo del vehículo en el estándar de navegación que es 0-360 grados, yendo en sentido horario.	"62.69"
<code>type</code>	Tipo del vehículo	"pkw"
<code>speed</code>	Velocidad del vehículo	"5.44"
<code>pos</code>	La posición de marcha del vehículo medida desde el inicio del carril actual	"12.14"
<code>lane</code>	Identificador de la pista actual	"144623880_0"
<code>slope</code>	La pendiente en grados. Es igual a la pendiente de la calle en la posición actual	"0.00"

Tabla 5.6: Descripción de los datos contenidos en archivo `FILE.xml`

Para procesar estos datos, utilizamos la herramienta `traceExporter.py` que provee SUMO. Los parámetros necesarios para la ejecución y obtención de los datos GPS son: `-fcd-input`

<archivo.xml>-gpsdat-output <destino.csv>. El archivo <destino.csv> contiene los datos resultantes y la estructura de dichos datos se presenta en la Tabla 5.7.

Id	Date-time	X	Y	Status	Speed km/h
46734_363_0	2018-11-06 06:01:15	11853.86	12489.55	70	30.456

Tabla 5.7: Estructura de los datos resultantes al ejecutar traceExporter.py

Considerando que para cada vehículo se obtiene la ruta completa y solo se necesita solo un tramo del recorrido, se procede a definir un área en torno a la intersección deseada. Se filtran todos los datos para obtener solo los datos dentro del area deseada, utilizando el script get_specific_gps_traces.py. Además se agrupan los datos por cada vehículo para tener la ruta de continuo y se transforma el valor de la velocidad de km/h a m/s.

La estructura de los datos queda finalmente como se presenta en la tabla 5.8.

X	Y	Speed m/s	Id	Date - time
11853.86	12489.55	8,46	46734_363_0	2018-11-06 06:01:15

Tabla 5.8: Estructura de datos procesados por get_specific_gps_traces.py

Las rutinas de procesamiento que se mencionan en este capítulo, se encuentran almacenadas en un repositorio en Github [38].

5.3. Resultados

Para la creación de una arquitectura de alerta temprana, se analizan cuatro arquitecturas de comunicación que se presentan en la Tabla 5.9, utilizando las herramientas presentadas en los capítulos anteriores y las evaluaremos según las métricas ya definidas. La finalidad es encontrar una combinación entre algoritmo y arquitectura, que permita alertar al ciclista sobre un potencial accidente y este, tenga tiempo suficiente para reaccionar mitigando o idealmente, evitando el potencial accidente.

Tabla 5.9: Arquitecturas de comunicación propuestas y sus puntos de procesamiento

Arquitectura	Punto de procesamiento
Vehículo - Bicicleta	Dispositivo en bicicleta
Vehículo - RSU - Bicicleta	RSU
Vehículo - RSU - LTE - Bicicleta	RSU
Vehículo - RSU - Server - Bicicleta	Server

Para determinar el alcance de las soluciones a evaluar, se utiliza un software de simulación, como es SUMO , OMNET++ que se integra con el Framework Veins [13]. En este software de simulación, se implementaron las cuatro arquitecturas de comunicación, en las cuales se integra el algoritmo de predicción de giros en el punto respectivo donde se realiza el procesamiento según lo que indica cada arquitectura como se observa en la Tabla 5.9. El escenario utilizado para probar las arquitecturas es una extracción de la ciudad Colonia en

Alemania, de donde se utiliza una intersección con una ciclovía como se muestra en la figura 5.1.

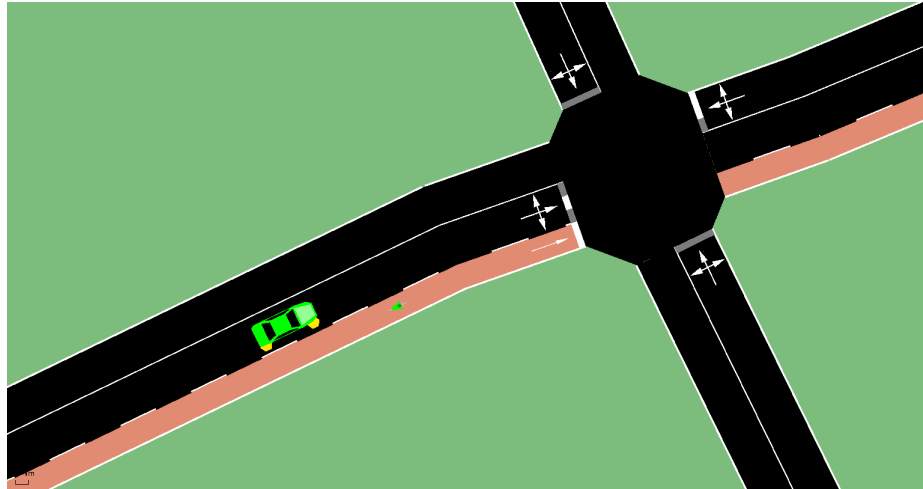


Figura 5.1: Intersección obtenida de Colonia, Alemania

Para evaluar las arquitecturas, se realizaron pruebas de rendimiento tomando en cuenta el retardo y pérdida de paquetes para analizar si dichas arquitecturas en conjunto con el algoritmo de predicción, logran predecir con tiempo suficiente el giro vehicular para alertar al ciclista y además, medir la eficacia del algoritmo en cuanto a la predicción del giro vehicular.

5.3.1. Descripción de la intersección

La intersección presentada en la figura 5.1, cuenta con dos calles de doble vía. La calle donde está la ciclovía simula una vía colectora y la calle que se une en la intersección, es una vía de servicio. En esa intersección, la vía de servicio tiene un disco pare como control de flujo y la vía colectora tiene la preferencia en el cruce. Según la definición entregada por la Ordenanza General de Urbanismo y Construcción (OGUC), las vías colectoras tienen una velocidad de diseño entre 40 km/h y 50 km/h.

5.4. Vehículo - Bicicleta

Para esta arquitectura, se considera que el vehículo se comunica directamente con un dispositivo integrado en la bicicleta utilizando el protocolo IEEE 802.11p como se muestra en la Figura 5.2 y por ende, el retardo generado por la red de comunicación es despreciable. El procesamiento del algoritmo de predicción se realiza en la bicicleta utilizando el dispositivo integrado para dicho propósito.

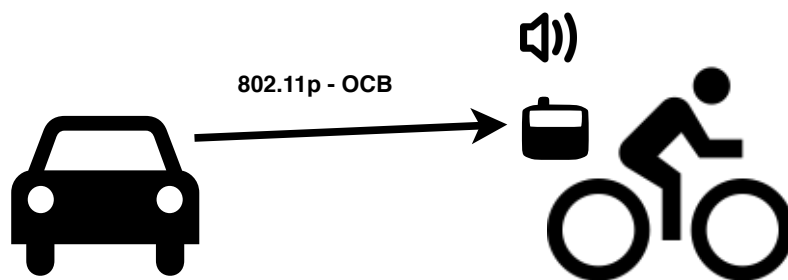


Figura 5.2: Arquitectura Vehículo-Ciclista

5.4.1. Time To Collision

Los parámetros con que se realizó la simulación se muestran en la Tabla 5.10. El objetivo de esta simulación fue encontrar la relación que tiene la velocidad máxima del vehículo con el TTC. Para cada velocidad dentro del rango a analizar, se realizaron 30 repeticiones, para encontrar el promedio y desviación estándar de los resultados entregados.

Tabla 5.10: Parámetros para el escenario vehículo bicicleta con diferentes velocidades

Parámetro	Valor
Protocolos de comunicación	802.11p
Procesamiento del algoritmo	Dispositivo Ciclista
Rango de velocidad máxima	[30 km/h, 31 km/h, 32 km/h, ..., 79 km/h]
Cantidad de velocidades analizadas	46
Repeticiones por cada velocidad	30
Total simulaciones escenario	1380

En el gráfico de la Figura 5.3, se observa como el TTC va aumentando hasta que se estabiliza en velocidades de 50 km/h y superiores, alcanzando un TTC de 2.2 segundos aproximadamente. La curva del gráfico, se comporta de esta manera ya que, a mayor velocidad, el conductor frena con mayor antelación a la esquina de calle. Considerando lo anterior, el algoritmo tiene mas tiempo para identificar este comportamiento de desaceleración y enviar la alerta correspondiente al ciclista.

Pérdida de Paquetes

Para este escenario, se decidió revisar el impacto que podría tener la pérdida de paquetes en el TTC, por lo que se realizó el experimento de inducir pérdida de paquetes a la arquitectura, utilizando los parámetros que se presentan en la tabla 5.11.

Los resultados obtenidos se reflejan en la figura 5.4 donde se puede ver que el TTC tiene un comportamiento estable y comienza a aumentar el TTC cuando el porcentaje de pérdida de paquetes supera el 90%. La razón de este comportamiento se debe a que el algoritmo de predicción comienza a fallar en cuanto a la precisión de su predicción como veremos en

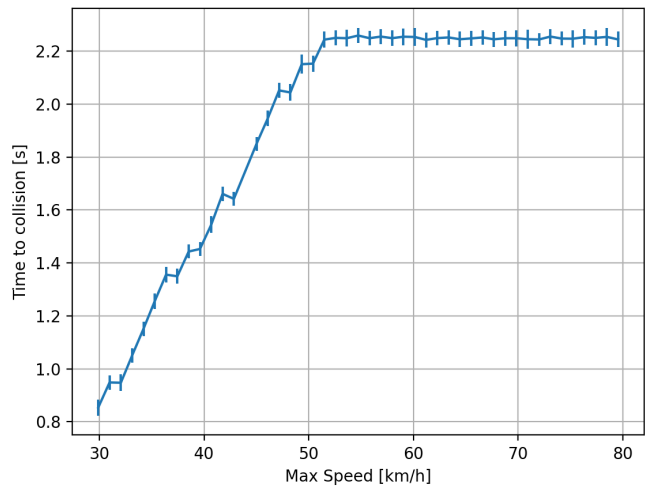


Figura 5.3: Tiempo para colisión según velocidad máxima, para arquitectura Vehículo - Bicicleta

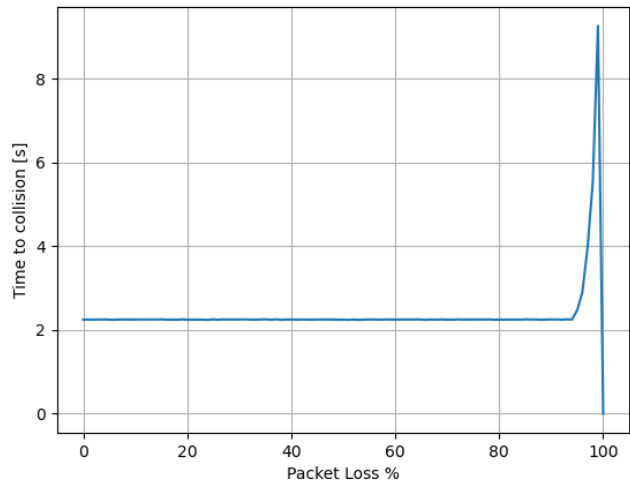


Figura 5.4: TTC por cada porcentaje de pérdida de paquetes, para arquitectura Vehículo - Bicicleta

la siguiente sección, y comienza a entregar resultados que no son fiables y por ende no se pueden considerar como útiles para el proceso de predicción.

Dado que el algoritmo de predicción se ejecuta en el dispositivo que tiene el ciclista, es que no se produce una disminución en cuanto al TTC ya que el mensaje de alerta no se pierde cuando el algoritmo detecta un posible giro y emite la alerta.

5.4.2. Precisión de la Predicción

Los parámetros con que se ejecutó la simulación para esta arquitectura, se presentan en la tabla 5.11 donde se puede ver que el algoritmo de predicción se ejecuta directamente en el dispositivo que tiene el ciclista. Dado esto, no hay riesgo de que el mensaje de alerta no llegue a su destino. Por otra parte, es posible que la cantidad de información recibida por el algoritmo no sea suficiente como para realizar una predicción correcta y en un instante útil para alertar al ciclista. La pérdida de paquetes está determinada por la capacidad del protocolo IEEE 802.11p.

Tabla 5.11: Parámetros para el escenario vehículo bicicleta para diferentes pérdida de paquetes

Parámetro	Valor
Protocolos de comunicación	802.11p
Procesamiento del algoritmo	Dispositivo Ciclista
Velocidad máxima	53 km/h
Rango de pérdida de paquetes	[0 %, 1 %, 2 %, 3 % ..., 100 %]
Cantidad de porcentajes de pérdida de paquetes analizados	101
Repeticiones por cada porcentaje de pérdida de paquetes	100
Total simulaciones escenario	10100

Para este escenario, se consideró un vehículo con un ciclista presentes en la intersección a estudiar y se repitieron las simulaciones cien veces para cada porcentaje de pérdida de paquetes. El objetivo de realizar la simulación de esta forma, es permitir obtener el impacto que tiene la pérdida de paquetes en el algoritmo de predicción, sin involucrar otros factores adicionales.

Los resultados obtenidos que se muestran en la Figura 5.5, indican que la pérdida de paquetes, comienza a afectar la precisión de la predicción cuando, la pérdida de paquetes supera un 80 %. Tomando en consideración que los mensajes se envían con una frecuencia de 10 mensaje por segundo, quiere decir que el algoritmo tolera la pérdida de 8 mensajes por segundo y aún así logra predecir correctamente el comportamiento del conductor.

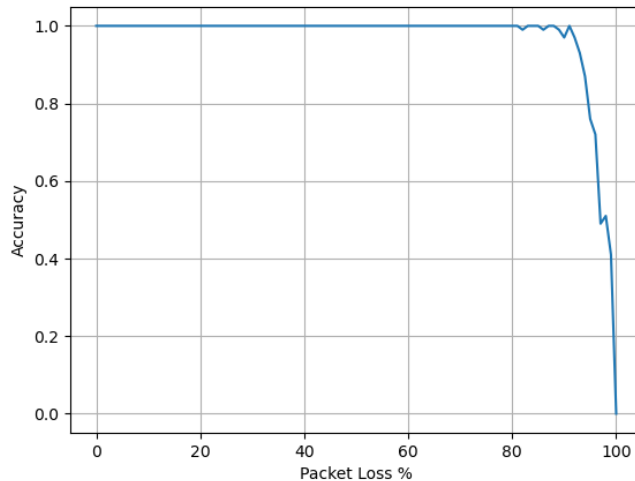


Figura 5.5: Precisión por cada porcentaje de pérdida de paquetes, para arquitectura Vehículo - Bicicleta

5.5. Vehículo - RSU - Bicicleta

Dentro de las comunicaciones inalámbricas vehiculares, se contempla la integración de las *Road Side Unit* (RSU), que son unidades de procesamiento que se colocan en las calles, con el fin de comunicar los vehículos con la infraestructura de las calles. Estas RSU, cuentan con la capacidad de conectarse utilizando el protocolo IEEE 802.11p y además pueden conectarse a internet. Para la arquitectura que se estudia en esta sección, se utiliza una RSU como centro de procesamiento del algoritmo de predicción.

Como se puede observar en la figura 5.6, el vehículo se comunica con la RSU utilizando el protocolo IEEE 802.11p y posteriormente la RSU se comunica con el dispositivo del ciclista utilizando el mismo protocolo IEEE 802.11p. La finalidad de considerar esta arquitectura, es mostrar la posibilidad de que el algoritmo de predicción se ejecute en estas RSU y envíe la alerta al ciclista en caso de ser necesario.

5.5.1. Time To Collision

Los parámetros con que se realizó la simulación se muestran en la tabla 5.12. En este caso, el procesamiento del algoritmo se realiza en la RSU y ésta se encarga de enviar el mensaje de alerta al teléfono móvil del ciclista en caso de que haya un posible giro y un potencial accidente.

El objetivo de esta simulación fue encontrar la relación que tiene la velocidad máxima con el TTC y determinar si es que realizar el procesamiento del algoritmo en la RSU podría afectar al desempeño del algoritmo. Para cada velocidad dentro del rango a analizar, se realizaron 30 repeticiones, para encontrar el promedio y desviación estándar de los resultados entregados.

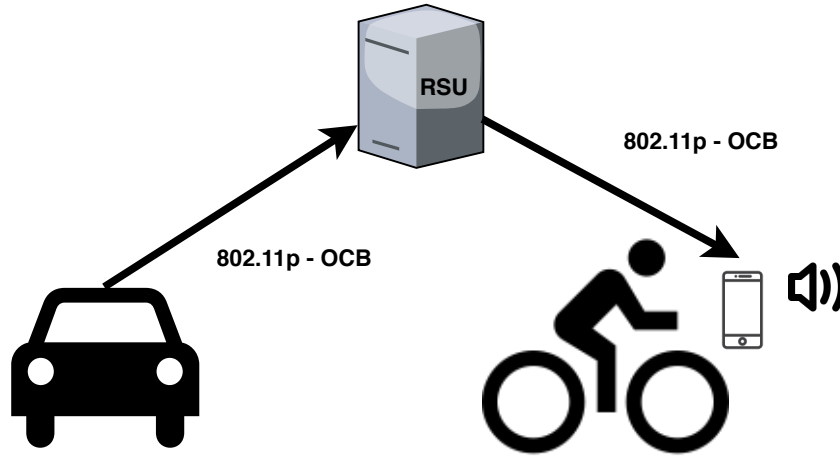


Figura 5.6: Arquitectura Vehículo - RSU - Ciclista

Tabla 5.12: Parámetros para el escenario vehículo - RSU - bicicleta con diferentes velocidades

Parámetro	Valor
Protocolos de comunicación	802.11p
Procesamiento del algoritmo	RSU
Rango de velocidad máxima	[30 km/h, 31 km/h, 32 km/h, ..., 79 km/h]
Cantidad de velocidades analizadas	46
Repeticiones por cada velocidad	30
Total simulaciones escenario	1380

El retardo producido por el protocolo IEEE 802.11p es despreciable para este escenario y como toda la transferencia de mensajes se realiza utilizando este protocolo, se desprecia el retardo. Dado lo anterior, es que el tiempo para colisión entregado por el algoritmo se asemeja al escenario anterior, como se puede apreciar en la figura 5.7.

Al ser el algoritmo de predicción el mismo y variar la arquitectura de comunicación con respecto al caso anterior, se puede observar que el comportamiento del TTC para ambos casos es similar. El TTC va aumentando a medida que aumenta la velocidad máxima hasta llegar a los 50 km/h donde alcanza un TTC de 2.2 segundos.

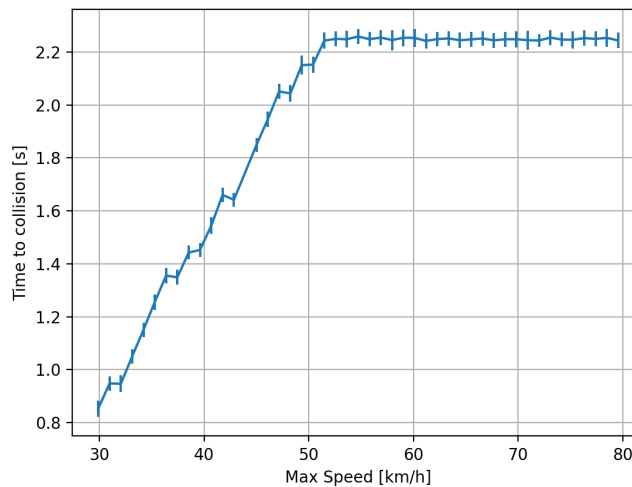


Figura 5.7: Tiempo para colisión según velocidad máxima, para arquitectura Vehículo - RSU - Bicicleta

Pérdida de Paquetes

Para verificar el impacto de la pérdida de paquetes en el TTC para esta arquitectura, se realizan varias simulaciones y los parámetros se presentan en la tabla 5.13

Se observa en la figura 5.8 que a medida que aumenta la pérdida de paquetes, comienza a disminuir el TTC. Esto se debe a que en esta arquitectura el procesamiento del algoritmo se realiza en la RSU para posteriormente enviar el mensaje de alerta al ciclista. La arquitectura de solución está diseñada para enviar varios mensajes de alerta al ciclista de forma secuencial para mitigar el impacto de la pérdida de paquetes y que de esta forma el ciclista reciba su mensaje de alerta. Considerando lo anterior, la declinación del TTC se debe a que el primer mensaje de alerta no llega al ciclista y un mensaje de alerta posterior logra cruzar la red y llegar al ciclista, por lo que se recibe finalmente la alerta pero de forma tardía. Al aumentar la pérdida de paquetes, se pierde una mayor cantidad de mensajes de alerta y por ende el mensaje que logra llegar al ciclista va mas tarde que el mensaje de alerta inicial.

Finalmente, al igual que en el caso anterior, la pérdida de paquetes es tanta que el algoritmo de predicción comienza a fallar y ya sobre el 90 % de pérdida de paquetes, deja de entregar

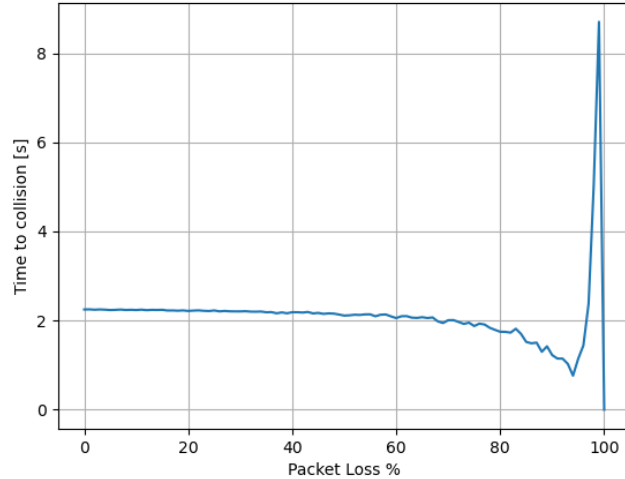


Figura 5.8: TTC por cada porcentaje de pérdida de paquetes, para arquitectura Vehículo - RSU - Bicicleta

resultados confiables de predicción.

5.5.2. Precisión de la Predicción

Los parámetros de simulación para esta prueba son como se describe en la tabla 5.13, donde destaca el hecho de que la ejecución del algoritmo de predicción se realiza en la RSU, el protocolo de comunicación para la arquitectura completa sigue siendo IEEE 802.11p y ahora el mensaje de alerta viaja por la arquitectura de red para llegar al ciclista.

Tabla 5.13: Parámetros para el escenario vehículo - RSU - bicicleta para diferentes pérdida de paquetes

Parámetro	Valor
Protocolos de comunicación	802.11p
Procesamiento del algoritmo	RSU
Velocidad máxima	53 km/h
Rango de pérdida de paquetes	[0 %, 1 %, 2 %, 3 %, 100 %]
Cantidad de porcentajes de pérdida de paquetes analizados	101
Repeticiones por cada porcentaje de pérdida de paquetes	100
Total simulaciones escenario	10100

Como se puede observar en la figura 5.9, la precisión del algoritmo de predicción tiene un comportamiento similar al de la arquitectura vehículo - bicicleta, donde se puede ver que la precisión se mantiene estable hasta el 80% de pérdida de paquetes para posteriormente presentar un declive en la precisión sobre el 90% de pérdida de paquetes.

El comportamiento de la precisión del algoritmo de giro para esta arquitectura sigue la misma lógica presentada en la sección de precisión de la predicción de la arquitectura vehículo - bicicleta.

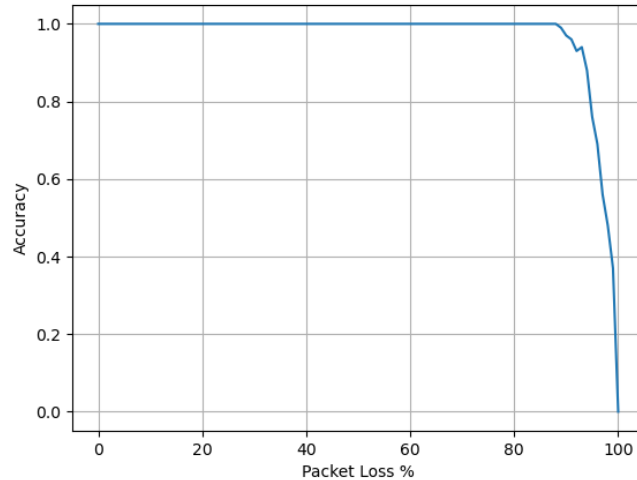


Figura 5.9: Precisión por cada porcentaje de pérdida de paquetes, para arquitectura Vehículo - RSU - Bicicleta

5.6. Vehículo - RSU - LTE - Bicicleta

Esta arquitectura es similar a la anterior como se puede observar en la figura 5.10 con la diferencia que ahora la RSU tiene la capacidad de enviar mensajes a través de LTE, por lo que la comunicación de la arquitectura de solución pasa por dos protocolos de comunicación distintos.

La comunicación entre el vehículo y la RSU sigue siendo por medio de IEEE 802.11p y la comunicación entre la RSU y el ciclista es por medio de LTE.

El algoritmo de predicción se ejecuta en la RSU y el mensaje de alerta se envía al ciclista a su teléfono móvil por medio de una red celular LTE.

5.6.1. Time To Collision

Los parámetros con que se realizó la simulación se muestran en la tabla 5.14. En este caso, se busca medir el impacto que tiene el utilizar LTE como red móvil intermedia para enviar el mensaje de alerta al ciclista.

Otro parámetro a considerar, es la inclusión de un rango de retardo que simula una baja calidad de servicio al utilizar LTE. El objetivo de esta inclusión de retardo es considerar esce-

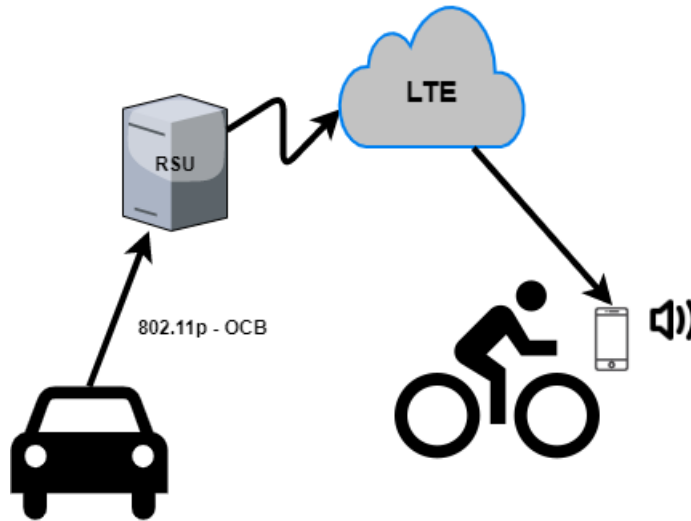


Figura 5.10: Arquitectura Vehículo - RSU - LTE - Ciclista

Tabla 5.14: Parámetros para el escenario vehículo - RSU - LTE - bicicleta con diferentes velocidades

Parámetro	Valor
Protocolos de comunicación	802.11p, LTE
Procesamiento del algoritmo	RSU
Rango de retardo añadido	100 ms - 200 ms
Rango de velocidad máxima	[30 km/h, 31 km/h, 32 km/h, ..., 79 km/h]
Cantidad de velocidades analizadas	46
Repeticiones por cada velocidad	30
Total simulaciones escenario	1380

narios donde el servicio entregado no sea bueno y comprobar el rendimiento de la arquitectura de alerta en condiciones realistas.

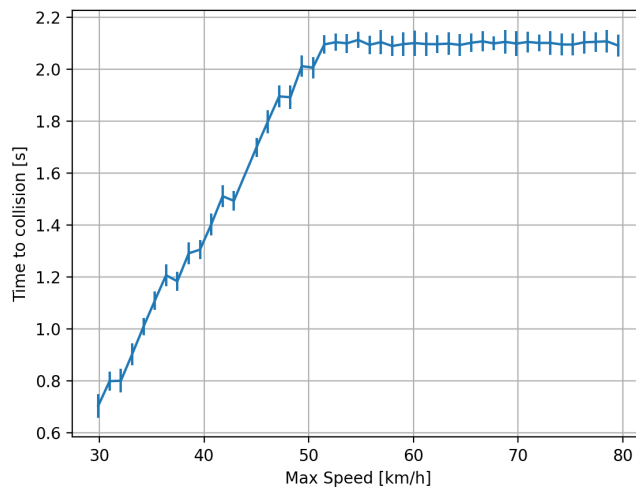


Figura 5.11: Tiempo para colisión según velocidad máxima, para arquitectura Vehículo - RSU - LTE - Bicicleta

Como se puede observar en la figura 5.11, el tiempo para colisión entregado por el algoritmo disminuye respecto a las arquitecturas anteriores, dejando el máximo en 2.1 segundos. Esta disminución se debe al retardo que se indujo en la simulación lo que nos lleva a concluir que el retardo afecta directamente al TTC entregado por la arquitectura de alerta.

Pérdida de Paquetes

Para verificar el impacto de la pérdida de paquetes en el TTC para esta arquitectura, se realizan varias simulaciones y sus parámetros se presentan en la tabla 5.15.

Como se observa en la figura 5.12, el comportamiento de la curva de TTC con respecto al porcentaje de pérdida de paquetes se asemeja a la curva de la figura 5.8 de la arquitectura Vehículo - RSU - Bicicleta. La diferencia que se presenta en este caso, es que el TTC es menor por el retardo que se ingresó a la arquitectura de solución por la inclusión del protocolo LTE encargado de enviar el mensaje de alerta al ciclista.

5.6.2. Precisión de la Predicción

Los parámetros de la simulación de pérdida de paquetes y la cantidad de repeticiones que se realizan se detallan en la tabla 5.15 en conjunto con los otros parámetros adicionales relevantes para la simulación.

Los resultados obtenidos de la inclusión de la pérdida de paquetes para la precisión del algoritmo de giro se observan en la figura 5.13. Se puede observar que la precisión del algoritmo

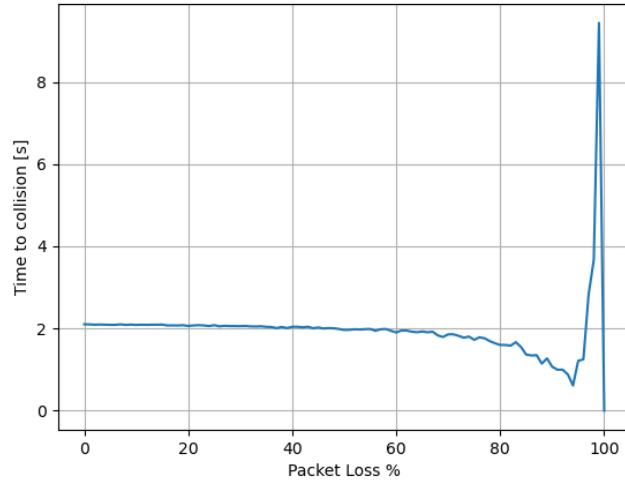


Figura 5.12: Precisión por cada porcentaje de pérdida de paquetes, para arquitectura Vehículo - RSU - LTE - Bicicleta

Tabla 5.15: Parámetros para el escenario vehículo - RSU - LTE - bicicleta para diferentes pérdida de paquetes

Parámetro	Valor
Protocolos de comunicación	802.11p, LTE
Procesamiento del algoritmo	RSU
Rango de retardo añadido	100 ms - 200 ms
Velocidad máxima	53 km/h
Rango de pérdida de paquetes	[0 %, 1 %, 2 %, 3 %, 100 %]
Cantidad de porcentajes de pérdida de paquetes analizados	101
Repeticiones por cada porcentaje de pérdida de paquetes	100
Total simulaciones escenario	10100

de giro frente a la pérdida de paquetes es muy similar a los resultados que se presentan en la figura 5.9, obtenidos de la arquitectura vehículo - RSU - ciclista.

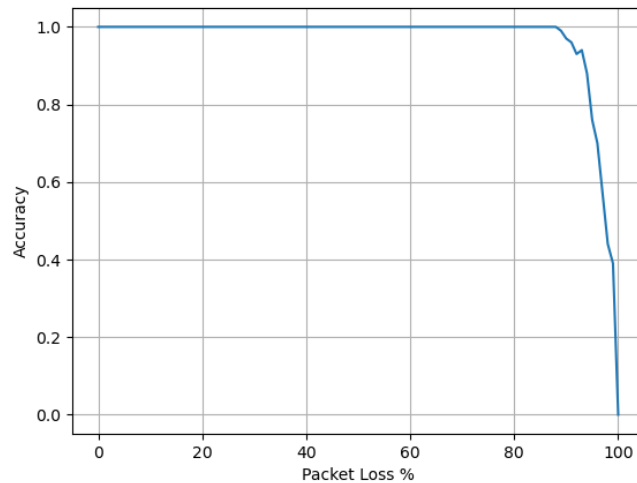


Figura 5.13: Cantidad de giros detectados por cada porcentaje de pérdida de paquetes, para arquitectura Vehículo - RSU - LTE - Bicicleta

Finalmente, se observa que sobre el 90% de pérdida de paquetes, comienza a bajar la cantidad de giros predichos correctamente, afectando la precisión de la predicción y por ende el rendimiento total de la arquitectura de alerta.

5.7. Vehículo - RSU - Server - Bicicleta

La arquitectura a analizar se muestra en la figura 5.14. En esta arquitectura, se agrega un componente adicional, siendo un servidor ubicado en la nube, que es el encargado de procesar el algoritmo de giro.

Para la comunicación en esta arquitectura, el vehículo envía sus mensajes por medio de IEEE 802.11p a la RSU, la cual se comunica con el servidor por medio de una red. Para efectos prácticos, no definimos que tipo de red se utilizará para la conexión entre la RSU y el servidor, ya que sin importar la red que se utilice, consideramos inducir un retardo a la comunicación entre la RSU y el servidor para poder abarcar un rango amplio de escenarios y de esta forma probar la arquitectura y su rendimiento sin que se requiera un escenario específico ni rendimientos óptimos. El objetivo de estas decisiones son acercar las simulaciones a situaciones realistas y de esta forma analizar la viabilidad de la arquitectura de solución en un escenario real.

Finalmente, los mensajes de alerta que envía el servidor al ciclista, llegan al teléfono móvil del ciclista por medio de LTE.

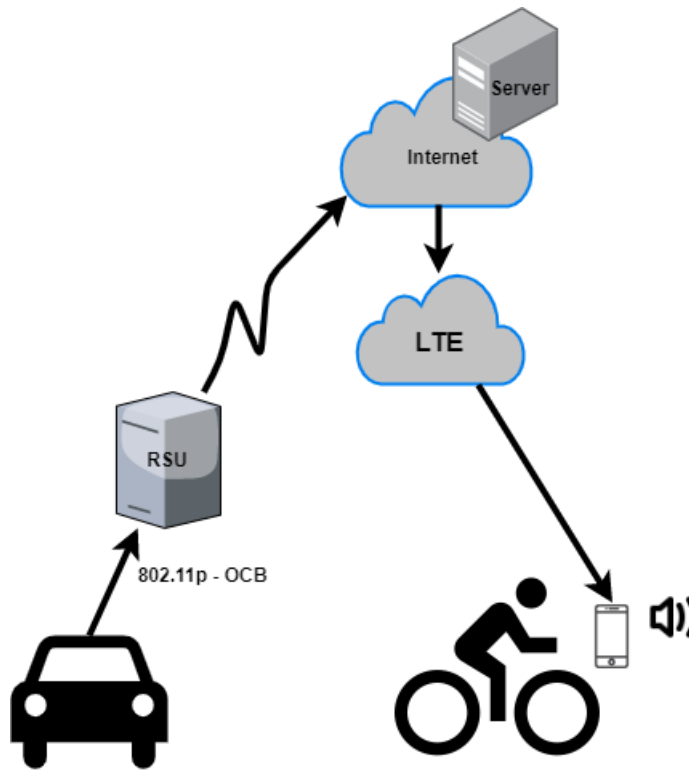


Figura 5.14: Arquitectura Vehículo - RSU - Server - Ciclista

5.7.1. Time To Collision

Los parámetros utilizados para este análisis de la arquitectura y su rendimiento en cuanto al TTC, se presentan en la tabla 5.16. Las diferencias que se presenta frente a las arquitecturas estudiadas anteriormente son: el procesamiento del algoritmo de predicción se realiza en un servidor en la nube, la red de comunicación incluye un paso por internet con un protocolo genérico y por último, se agrega un retardo a la arquitectura de solución en dos puntos de la arquitectura dado el uso de la red de internet y LTE.

Tabla 5.16: Parámetros para el escenario vehículo - RSU - Server - bicicleta con diferentes velocidades

Parámetro	Valor
Protocolos de comunicación	802.11p, LTE
Procesamiento del algoritmo	Server
Rango de retardo añadido	200 ms - 400 ms
Rango de velocidad máxima	[30 km/h, 31 km/h, 32 km/h, ..., 79 km/h]
Cantidad de velocidades analizadas	46
Repeticiones por cada velocidad	30
Total simulaciones escenario	1380

Los resultados de las simulaciones realizadas se muestran en la figura 5.15. Se observa nuevamente que el comportamiento del TTC sigue el mismo patrón presente en las arquitecturas y simulaciones anteriores, variando en este caso el TTC máximo que llega hasta

los 1.9 segundos. Se puede apreciar entonces que el retardo que se induce en la red afecta directamente el TTC entregado por la arquitectura de solución, ya que al compararlo con los resultados obtenidos para la arquitectura vehículo - bicicleta, se puede ver una disminución aproximada de 300 milisegundos.

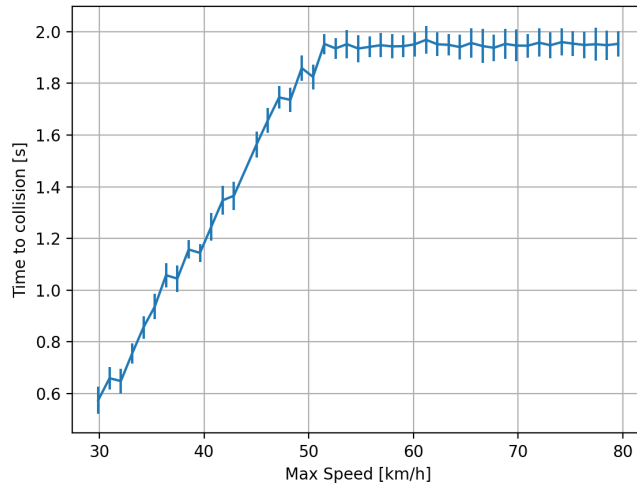


Figura 5.15: Tiempo para colisión según velocidad máxima, para arquitectura Vehículo - RSU - Servidor - Bicicleta

Pérdida de Paquetes

Los parámetros de la simulación para la pérdida de paquetes y su impacto en el TTC se muestran en la tabla 5.17. Cabe destacar que el algoritmo de predicción se ejecuta en el servidor y además se indujo retardo a la arquitectura para simular condiciones poco favorables de los canales de comunicación.

Se observa en la figura 5.16 que el comportamiento de la curva de TTC con respecto a la pérdida de paquetes es similar a las arquitecturas que deben enviar los mensajes de alerta al ciclista por una de comunicación. Es por esto que a medida que aumenta la pérdida de paquetes el TTC va disminuyendo hasta que el porcentaje de pérdida de paquetes es muy grande y las predicciones dejan de ser fiables.

Adicionalmente se puede ver que el TTC es menor que los TTC obtenidos para las arquitecturas expuestas previamente. Como ya se infiere, esto es debido a que el retardo afecta directamente al TTC y como esta arquitectura es la que tiene el mayor retardo inducido, presenta un TTC menor.

5.7.2. Precisión de la Predicción

Los parámetros de la simulación de pérdida de paquetes y la cantidad de repeticiones que se realizan se detallan en la tabla 5.17. Los cambios con respecto a las arquitecturas

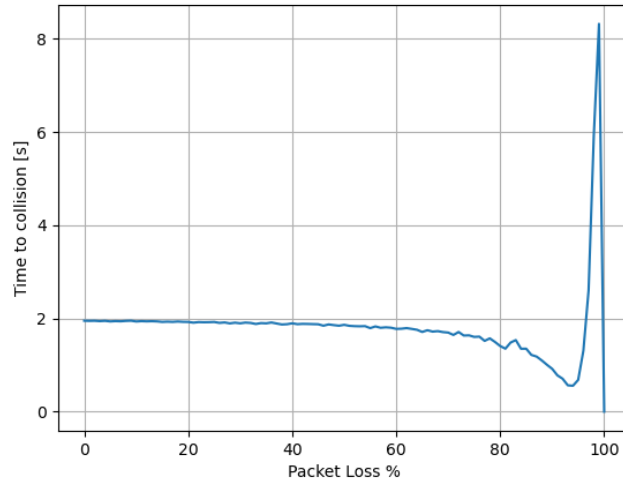


Figura 5.16: TTC por cada porcentaje de pérdida de paquetes, para arquitectura Vehículo - RSU - Servidor - Bicicleta

anteriores corresponden a que el algoritmo se procesa en el servidor y se induce un retardo mayor para la simulación de las redes de comunicación.

Tabla 5.17: Parámetros para el escenario vehículo - RSU - LTE - bicicleta para diferentes pérdida de paquetes

Parámetro	Valor
Protocolos de comunicación	802.11p, LTE
Procesamiento del algoritmo	Server
Rango de retardo añadido	200 ms - 400 ms
Velocidad máxima	53 km/h
Rango de pérdida de paquetes	[0 %, 1 %, 2 %, 3 %, 100 %]
Cantidad de porcentajes de pérdida de paquetes analizados	101
Repeticiones por cada porcentaje de pérdida de paquetes	100
Total simulaciones escenario	10100

Los resultados de analizar la precisión para cada porcentaje de pérdida de paquetes se muestra en la figura 5.17. Como se puede observar, los resultados obtenidos siguen el mismo comportamiento que en las arquitecturas anteriores. Esto se debe principalmente a que el algoritmo de predicción funciona de forma transparente con respecto a la arquitectura en la que se montó en cuanto a la precisión de sus predicciones.

Es por lo mencionado anteriormente que el algoritmo presenta buena precisión hasta sobre el 80 % de pérdida de paquetes. En este punto comienza a fallar en cuanto a la predicción y por ende disminuye la precisión de las predicciones de la arquitectura de solución.

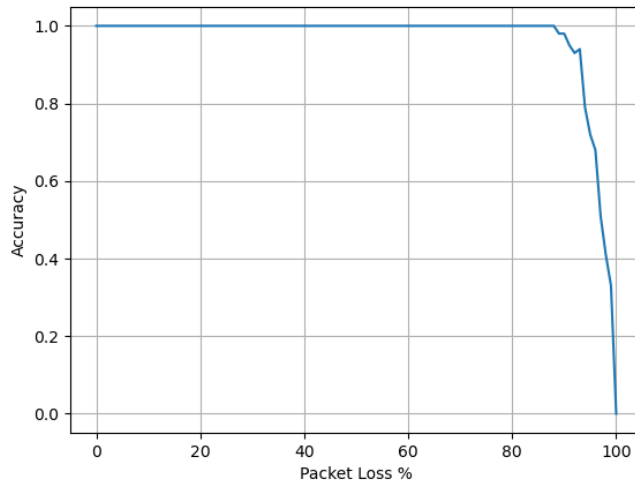


Figura 5.17: Cantidad de giros detectados por cada porcentaje de pérdida de paquetes, para arquitectura Vehículo - RSU - Servidor - Bicicleta

5.8. Precisión según cantidad de vehículos

En esta sección se muestran los resultados que se obtuvieron de probar la precisión del algoritmo de giro frente a diferentes cantidades de vehículos presentes en la calle simultáneamente. Esta prueba es de interés porque según se presenta en [23], las caravanas de vehículos que se forman en una calle impactan en la capacidad del algoritmo de predicción.



Figura 5.18: Escenario con cinco vehículos presentes en simultáneo en la vía

Al producirse caravanas de vehículos, el algoritmo de predicción comienza a presentar fallos en la predicción de giros como se puede observar en la figura 5.19.

Para realizar estas pruebas, se utilizó la arquitectura [vehículo - RSU - server - ciclista] y el mismo escenario con el que se probaron las diferentes arquitecturas de comunicación. Por cada iteración se generaron 300 rutas aleatorias que se clasificaron de la siguiente forma:

- Giro a la derecha
- Continuar recto

Para poder controlar la cantidad de vehículos que había presente en la calle previa a la intersección se espaciaron los tiempos en que los vehículos ingresan al escenario, para que se cumpliera para las 300 rutas la restricción de la cantidad de vehículos presentes en la calle.

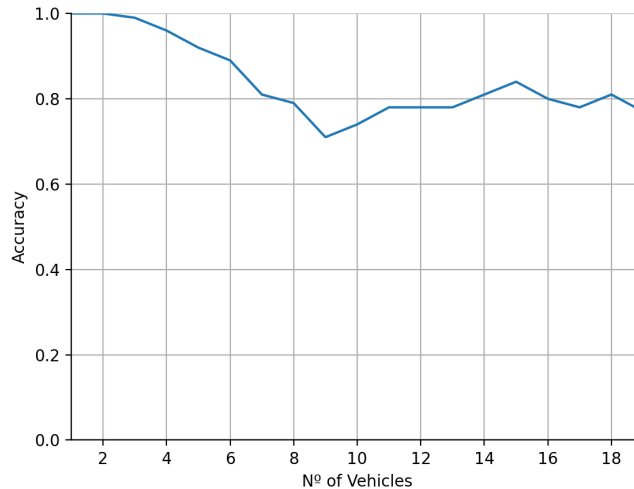


Figura 5.19: Precisión según la cantidad de vehículos

Los resultados obtenidos de estas simulaciones se presentan en las tablas 5.18 y 5.19 donde:

- True Positives: Comportamientos de giro a la derecha detectados correctamente
- False Positives: Comportamientos de giro a la derecha detectados incorrectamente
- True Negatives: Comportamientos de seguir recto detectados correctamente
- False Negatives: Comportamientos de seguir recto detectados incorrectamente

Se puede observar que la precisión es muy buena mientras hay pocos vehículos presentes en la calle en simultáneo y la precisión va disminuyendo a medida que van aumentando los vehículos presentes en simultáneo en la calle.

Otro punto importante que se obtiene de las tablas, es la ausencia de falsos negativos. Esto implica que el algoritmo siempre detecta giros a la derecha y los casos en los que falla en su predicción, es cuando un vehículo que va a seguir recto lo detecta como un posible giro hacia la derecha. Esto es muy bueno para nuestra arquitectura de solución, ya que no habría casos en que el algoritmo no alerte al ciclista de un potencial giro por confundirlo con un vehículo que pretende seguir recto.

Siguiendo la misma línea y según se presenta en [23], este comportamiento se debe a que cuando se forman caravanas, si el vehículo que va primero desacelera para girar a la derecha en la intersección, los vehículos que le siguen van a desacelerar para no chocar con los vehículos que están adelante sin importar si es que van a seguir recto o girar a la derecha en la intersección. Es por lo mencionado anteriormente que el algoritmo confunde vehículos

que van a seguir recto con potenciales giros.

Tabla 5.18: Precisión y resultados de predicción por cantidad de vehículos para **300 rutas**

Cantidad de Vehículos	1	2	3	4	5	6	7	8	9	10
True Positives	147	153	138	165	144	151	145	156	139	151
False Positives	0	0	3	11	24	32	58	62	88	79
True Negatives	153	147	159	124	132	117	97	82	73	70
False Negatives	0	0	0	0	0	0	0	0	0	0
Accuracy	1	1	0.99	0.96	0.92	0.89	0.81	0.79	0.71	0.74

Tabla 5.19: Precisión y resultados de predicción por cantidad de vehículos para **300 rutas**

Cantidad de Vehículos	11	12	13	14	15	16	17	18	19
True Positives	158	141	139	158	163	155	141	164	152
False Positives	65	65	65	57	48	61	66	57	68
True Negatives	77	94	96	85	89	94	93	79	80
False Negatives	0	0	0	0	0	0	0	0	0
Accuracy	0.78	0.78	0.78	0.81	0.84	0.80	0.78	0.81	0.77

5.9. Discusión Resultados

5.9.1. Time To Collision

Como se observó en las diferentes arquitecturas, el retardo tiene un impacto directo en el TTC otorgado por el algoritmo de predicción de giros. Esto se debe a que el retardo producido por el traspaso de mensajes por los canales de comunicación afectan directamente al tiempo que se demora en procesar el algoritmo cada paso del modelo para calcular la probabilidad del posible giro. Es por esto que para las arquitecturas que se manejan solo con el protocolo IEEE 802.11p se obtienen los valores mas altos de TTC ya que el retardo que se genera por el canal es mínimo y despreciable.

Medir el retardo promedio de internet no es una tarea fácil ya que la cantidad de factores que influyen es muy grande y la variabilidad de los escenarios no permite tomar un "promedio" que tenga alguna utilidad relevante. Es por esto que se decide proponer un rango de retardo que sería un caso relativamente malo si consideramos que el retardo se ve afectado directamente por la distancia entre los dispositivos a comunicar. En este caso, consideramos a Santiago de Chile como punto inicial y comparamos el retardo que hay con servidores en Brasil o Estados Unidos para determinar cuanto se podrían demorar los paquetes en llegar en un caso favorable de conexión. Tomando que en promedio estas pruebas de retardo no superaron los 80 milisegundos, se decidió utilizar un rango entre 100 a 200 milisegundos para simular un canal de comunicación deficiente para ver como afectaría al algoritmo de predicción y de esta forma abarcar un amplio rango de escenarios donde cualquier retardo menor a 100 milisegundos podría ser una mejora dentro de los escenarios que estudiamos en las simulaciones.

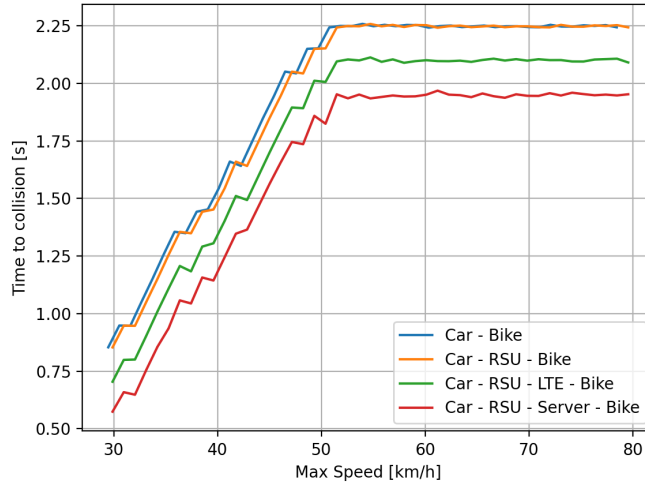


Figura 5.20: Tiempo para colisión según velocidad máxima, para las cuatro arquitecturas presentadas

Como se puede ver en la Figura 5.20, el retardo afecta directamente al TTC del algoritmo por lo que en las arquitecturas donde solo hay comunicación utilizando IEEE 802.11p se alcanzan los mejores tiempos de predicción llegando a los 2.2 segundos. Por otro lado, la arquitectura que utiliza LTE para enviar la alerta al ciclista alcanza máximos de 2.1 segundos y finalmente, la arquitectura que debe pasar por el servidor llega a máximos de 1.9 segundos aproximadamente.

Dado todo esto, se puede observar que mientras mas conexiones se deban realizar con internet para manejar los mensajes ya sea de los vehículos o de alerta para los ciclistas, el TTC disminuirá.

5.9.2. Velocidad Máxima

La velocidad máxima a la que circulan los vehículos también impacta en el algoritmo de predicción ya que se considera la desaceleración como un factor dentro del modelo para poder determinar si el vehículo va a doblar en la esquina o no. Como se puede apreciar en las figuras de TTC por velocidad máxima, el TTC va aumentando junto con la velocidad máxima hasta llegar a los 50 km/h donde posterior a dicha velocidad, el TTC ya no aumenta manteniéndose entre los 2.3 y 1.9 segundos dependiendo de la arquitectura estudiada. Este factor de velocidad es importante, ya que si los vehículos circulan a una velocidad baja, el TTC será menor y podría no ser suficiente para alertar al ciclista y que este pueda reaccionar a tiempo para prevenir o mitigar la colisión.

En la Figura 5.21, se agregan líneas horizontales que representan el tiempo necesario que necesitan los ciclistas para frenar, según la velocidad con la que se desplazan por la ciclovía. Estos TTC se consideran como el tiempo total requerido por los ciclistas para frenar como se presenta en la Tabla 4.3. Es importante considerar que los datos de la tabla, son para un

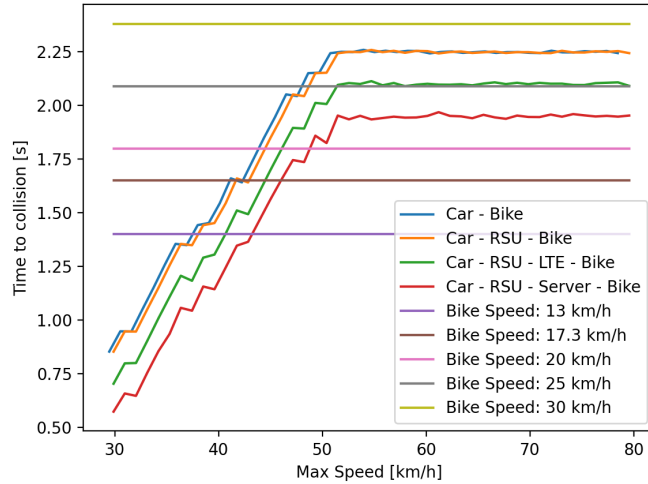


Figura 5.21: TTC - velocidad máxima y TTC para bicicletas

escenario específico, donde la ciclovía es plana, y el concreto está seco. Para otros escenarios donde el concreto está húmedo y hay inclinación, los tiempos de frenado de los ciclistas puede cambiar.

Como ya hemos analizado, los vehículos alcanzan el TTC máximo sobre los 50 km/h, lo que limita los escenarios donde se podría aplicar la arquitectura de solución, ya que en algunas calles urbanas, las velocidades de conducción disminuyen, lo que generará menor TTC para los vehículos, y si los ciclistas circulan a mayor velocidad, requerirán más tiempo para poder realizar maniobras evasivas. A pesar de lo anterior, si consideramos la arquitectura vehículo - ciclista, que tiene los mejores resultados de TTC, podría ser una arquitectura de solución viable en una mayor cantidad de escenarios, ya que las bajas latencias, permitirían alertar con tiempo suficiente a una mayor cantidad de ciclistas que viajen a mayor velocidad.

Finalmente, el escenario favorable, consiste en avenidas donde la velocidad de circulación se acerca a los 50 km/h, y los ciclistas viajan a menos de 25 km/h por la ciclovía, ya que en este escenario se alcanzaría la mayor cobertura, permitiendo que la arquitectura de solución envíe alertas con TTC suficiente para que los ciclistas puedan frenar.

5.9.3. Pérdida de Paquetes

La pérdida de paquetes tuvo un impacto similar para todas las arquitecturas, en donde el algoritmo comenzó a presentar fallas en la predicción en torno al 90 % de pérdida de paquetes.

Estos resultados son bastante prometedores ya que quiere decir que el algoritmo es robusto frente a la pérdida de paquetes, lo que permite enfocar las energías en mejorar los tiempos de predicción y disminuir el retardo dejando menos recursos y esfuerzos a la mitigación de la pérdida de paquetes. Además, si los canales de comunicación presentaran bajo retardo pero algún porcentaje de pérdida de paquetes, el algoritmo seguiría funcionando sin mayores

complicaciones, lo que amplía las posibilidades frente a los diferentes escenarios donde podría haber una baja calidad de señal con una cantidad considerable de pérdida de paquetes, siempre y cuando el retardo no aumente considerablemente.

Como se puede observar en los gráficos de TTC por porcentaje de pérdida de paquetes, el TTC se ve afectado para los escenarios donde los mensajes de alerta deben ser enviados por alguna red para poder alertar al ciclista. Considerando esto, el caso ideal es cuando el procesamiento del algoritmo se realiza en el nodo mismo, que en este caso sería el dispositivo del ciclista. Finalmente, sobre el 90 % de pérdida de paquetes, el TTC entregado no es fiable ya que la precisión del algoritmo disminuye mucho como para poder considerar los datos entregados como confiables.

Conclusión

Como se presentó anteriormente, el aumento del uso de la bicicleta como medio de transporte ha traído nuevos desafíos para todos los actores de las vías de tránsito, teniendo que convivir y adaptarse a las nuevas formas de desplazarse tanto en Chile como en el mundo.

A lo largo del presente trabajo se abordó el problema de cómo prevenir los accidentes entre vehículos y ciclistas en las intersecciones de calle, planteando la necesidad de implementar sistemas de alerta temprana con el propósito de alertar al ciclista de las potenciales colisiones con vehículos que desean girar en una intersección de calle y de esta forma ayudar al ciclista a prevenir o mitigar dichas colisiones.

En este trabajo se utilizaron datos obtenidos del escenario TAPAS Cologne que describen el tráfico de un día completo de la ciudad Colonia en Alemania. Con estos datos se seleccionó una intersección de interés donde hubiese flujo vehicular para realizar las pruebas de eficacia del algoritmo de predicción de giros y de rendimiento de las arquitecturas de comunicación estudiadas en este trabajo. Además, se realizó la implementación del algoritmo de giro dentro de las diferentes arquitecturas de comunicación las cuales se simularon utilizando VEINS, SUMO y OMNET++ para realizar simulaciones que permitieron obtener resultados de rendimiento tanto del algoritmo como de las arquitecturas de comunicación.

Como conclusión del trabajo de tesis realizado, se puede mencionar que para escenarios donde la velocidad de circulación se acerca a los 50 km/h y los ciclistas circulan a menos de 26 km/h, las arquitecturas logran alertar en la mayoría de los casos a los ciclistas, otorgándoles tiempo suficiente para reaccionar. Por otra parte, para los otros escenarios donde las velocidades de circulación vehicular son menores, se debe continuar analizando y mejorando tanto el algoritmo de predicción como las arquitecturas de comunicación, ya que en estos escenarios, solo las arquitecturas que se comunican utilizando IEEE 802.11p exclusivamente, presentan TTC útiles para los ciclistas.

Junto a lo anterior, se muestra que las diferentes arquitecturas junto con el algoritmo de predicción tienen buena tolerancia a la pérdida de paquetes, ya que resisten sobre un 80 % de pérdida de paquetes antes de que se vea afectada la precisión del algoritmo de giros, por lo que este algoritmo podría ser útil en escenarios adversos donde la pérdida de paquetes está presente en la red.

Además, el algoritmo de predicción tuvo un buen desempeño en cuanto a la precisión para detectar giros vehiculares, logrando valores sobre un 70 % para los diferentes escenarios de densidad vehicular, mostrando una precisión superior a 90 % cuando hay cinco o menos

vehículos presentes en la calle de forma simultánea. Dentro del mismo marco, el algoritmo presentó un muy buen desempeño detectando todos los giros vehiculares hacia la derecha sin presentar falsos negativos. Esto cobra mayor relevancia si es que consideramos que al no haber giros sin detectar, permite dentro de algunos escenarios que el ciclista reciba siempre una alerta previniéndole de un potencial accidente.

Sobre los objetivos planteados, se puede concluir que se logró cumplir con lo planteado, ya que se pudo estudiar, diseñar, implementar y evaluar una arquitectura de alerta temprana para ciclistas. Esto se puede observar en los diferentes capítulos donde se construye la arquitectura de la solución en cada etapa, se analizan los diferentes componentes de la arquitectura y finalmente se estudia y evalúa la arquitectura de solución de forma global, determinando las restricciones y los alcances de dicha solución.

Por lo tanto considerando todo lo ya presentado, es posible dar respuesta a la hipótesis y problema a resolver planteados al comienzo de este documento, demostrando la factibilidad para escenarios específicos, de crear un sistema de comunicación, en conjunto con un algoritmo de predicción de giros vehiculares hacia la derecha en intersecciones, que permita alertar a los ciclistas con tiempo suficiente para que reaccionen y puedan de esta forma mitigar o prevenir un accidente.

5.10. Trabajo Futuro

5.10.1. Arquitecturas con 5G

Con el fin de mejorar las arquitecturas de comunicación para la arquitectura de alerta para ciclistas, se propone estudiar el protocolo de comunicación 5G ya que en la literatura se propone como un protocolo que permite la integración de comunicación vehicular con otros agentes presentes en las calles. Un ejemplo del estudio de 5G para la integración vehicular con otros agentes (V2X), se presenta en [14].

5.10.2. Otros Algoritmos de predicción

Encontrar algoritmos de predicción que ofrezcan buenos resultados y otorguen mas opciones a los ciclistas es fundamental para disminuir los accidentes con ciclistas involucrados. Analizar series temporales y rutas ya recorridas en conjunto con aprendizaje profundo, es una posible alternativa como se presenta en [43]. Analizar las rutas por donde circulan los vehículos y las intenciones naturales de los conductores es otra forma de predecir el comportamiento de los vehículos en intersecciones como se presenta en [44] utilizando redes neuronales recurrentes. Otra propuesta se presenta en [33] donde proponen un algoritmo basado en la simulación de Monte Carlo para determinar las potenciales colisiones y predecir los comportamientos vehiculares.

5.10.3. Métodos de alerta para ciclistas

Encontrar formas de alertar a ciclistas de potenciales accidentes es una buena manera de disminuir las muertes y accidentes con usuarios vulnerables de las calles, que en este caso son ciclistas. Una propuesta para un sistema de alerta de accidentes para ciclovías se presenta en [42] el cuál podría adaptarse para enviar alertas a los ciclistas de los potenciales accidentes.

Sistemas de prevención pasiva para ciclistas

Además de los sistemas activos de prevención para ciclistas, es interesante considerar el impacto que podría tener la integración conjunta de los sistemas estudiados con las señaléticas de tránsito, tales como discos pare, preferencia a ciclistas y entre otras.

Bibliografía

- [1] Sassan Ahmadi. Introduction to LTE-advanced. In *LTE-Advanced*, pages 1–27. Elsevier, 2014.
- [2] A. Bender, G. Agamennoni, J. R. Ward, S. Worrall, and E. M. Nebot. An unsupervised approach for inferring driver behavior from naturalistic driving data. *IEEE Transactions on Intelligent Transportation Systems*, 16(6):3325–3336, Dec 2015.
- [3] A. Bender, J. R. Ward, S. Worrall, and E. M. Nebot. Predicting driver intent from models of naturalistic driving. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pages 1609–1615, Sept 2015.
- [4] Silvia Bernardi and Federico Rupi. An analysis of bicycle travel speed and disturbances on off-street and on-street facilities. *Transportation Research Procedia*, 5:82–94, 2015.
- [5] H. Berndt and K. Dietmayer. Driver intention inference with vehicle onboard sensors. In *2009 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, pages 102–107, Nov 2009.
- [6] Kim Buchholtz and Theresa Burgess. An evaluation of bicycle-specific agility and reaction times in mountain bikers and road cyclists. *South African Journal of Sports Medicine*, 32(1):1–5, September 2020.
- [7] Shinko Yuanhsien Cheng and M. M. Trivedi. Turn-intent analysis using body pose for intelligent driver assistance. *IEEE Pervasive Computing*, 5(4):28–37, Oct 2006.
- [8] CONASET. Comisión Nacional de Seguridad de Tránsito. "<https://www.conaset.cl/ciclistas/>, 2018. (Visitado 21-11-2021).
- [9] K. Dhondge, S. Song, B. Choi, and H. Park. Wifihonk: Smartphone-based beacon stuffed wifi car2x-communication system for vulnerable road user safety. In *2014 IEEE 79th Vehicular Technology Conference (VTC Spring)*, pages 1–5, May 2014.
- [10] Eco-Counter. Worldwide Cycling Index. <https://www.eco-counter.com/2019-worldwide-cycling-index/>, 2019. (Visitado 21-11-2021).
- [11] Stephan Eichler. Performance evaluation of the IEEE 802.11p WAVE communication standard. In *2007 IEEE 66th Vehicular Technology Conference*. IEEE, September 2007.

- [12] Jenny Eriksson, Åsa Forsman, Anna Niska, Susanne Gustafsson, and Gunilla Sörensen. An analysis of cyclists' speed at combined pedestrian and cycle paths. *Traffic Injury Prevention*, 20(sup3):56–61, September 2019.
- [13] Mohammad Jan Haidari and Zeki Yetgin. Veins based studies for vehicular ad hoc networks. In *2019 International Artificial Intelligence and Data Processing Symposium (IDAP)*. IEEE, September 2019.
- [14] Shimaa A. Abdel Hakeem, Anar A. Hady, and HyungWon Kim. 5g-v2x: standardization, architecture, use cases, network-slicing, and edge-computing. *Wireless Networks*, 26(8):6015–6041, July 2020.
- [15] U. Hernandez-Jayo, J. Perez, I. de-la Iglesia, and R. Carballedo. Cs4vr: Remote monitoring and warning system for vulnerable road. In *2016 13th International Conference on Remote Engineering and Virtual Instrumentation (REV)*, pages 153–158, Feb 2016.
- [16] Unai Hernandez-Jayo, Idoia De la Iglesia, and Jagoba Perez. V-alert: Description and validation of a vulnerable road user alert system in the framework of a smart city. *Sensors*, 15(8):18480–18505, jul 2015.
- [17] Richard Van Der Horst and Jeroen Hogema. 6th ictet workshop salzburg proceedings 1 time-to-collision and collision avoidance systems.
- [18] Sustainability Info. Discrete event simulator. "<https://sustainabilityinfo.com/transportation/what-is-the-average-speed-for-urban-cycling/>", 2019. (Visitado 21-11-2021).
- [19] ITARDA. Institute for Traffic Accident Research and Data Analysis. "<http://www.itarda.or.jp/english/>", 2018. (Visitado 21-11-2021).
- [20] Ruobing Jiang and Yanmin Zhu. Wireless access in vehicular environment. In *Encyclopedia of Wireless Networks*, pages 1–5. Springer International Publishing, 2019.
- [21] Stéphanie Lefèvre, Dizan Vasquez, and Christian Laugier. A survey on motion prediction and risk assessment for intelligent vehicles. *ROBOMECH Journal*, 1(1), jul 2014.
- [22] M. Liebner, M. Baumann, F. Klanner, and C. Stiller. Driver intent inference at urban intersections using the intelligent driver model. In *2012 IEEE Intelligent Vehicles Symposium*, pages 1162–1167, June 2012.
- [23] M. Liebner, F. Klanner, M. Baumann, C. Ruhhammer, and C. Stiller. Velocity-based driver intent inference at urban intersections in the presence of preceding vehicles. *IEEE Intelligent Transportation Systems Magazine*, 5(2):10–21, Summer 2013.
- [24] M. Liebner, C. Ruhhammer, F. Klanner, and C. Stiller. Generic driver intent inference based on parametric models. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pages 268–275, Oct 2013.
- [25] NHTSA. National Highway Traffic Safety Administration. "<https://www.nhtsa.gov/>",

2018. (Visitado 21-11-2021).
- [26] NHTSA. National highway traffic safety administration, vehicle to vehicle communication. "<https://www.nhtsa.gov/technology-innovation/vehicle-vehicle-communication>", 2018. (Visitado 21-11-2021).
- [27] OMNeT++. Discrete event simulator. <https://omnetpp.org/>, 2018. (Visitado 21-11-2021).
- [28] AASHTO Task Force on Geometric Design. Aashto guide for the development of bicycle facilities. <https://njdotlocalaidrc.com/perch/resources/aashto-gbf-4-2012-bicycle.pdf>, 2012. (Visitado 21-11-2021).
- [29] Andreas Rutgersson. A study of cyclists' need for an intelligent transport system (its) (master's thesis). Master's thesis, Chalmers University of Technology, Göteborg, Sweden, 2013.
- [30] Mahmoud Saffarzadeh, Navid Nadimi, Saber Naseralavi, and Amir Reza Mamdoohi. A general formulation for time-to-collision safety indicator. *Proceedings of the Institution of Civil Engineers - Transport*, 166(5):294–304, October 2013.
- [31] J. Santa, P. J. Fernández, and M. A. Zamora. Cooperative its for two-wheel vehicles to improve safety on roads. In *2016 IEEE Vehicular Networking Conference (VNC)*, pages 1–4, Dec 2016.
- [32] Peter Schantz. Distance, duration, and velocity in cycle commuting: Analyses of relations and determinants of velocity. *International Journal of Environmental Research and Public Health*, 14(10):1166, October 2017.
- [33] Chuanliang Shen, Shan Zhang, Zhenhai Gao, Binyu Zhou, Wei Su, and Hongyu Hu. Study on a right-turning intelligent vehicle collision warning and avoidance algorithm based on monte carlo simulation. *Journal of Advanced Transportation*, 2020:1–11, August 2020.
- [34] SUMO. Simulation of urban mobility. <http://sumo.dlr.de/index.html>, 2018. (Visitado 21-11-2021).
- [35] A. Tamke, T. Dang, and G. Breuel. A flexible method for criticality assessment in driver assistance systems. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 697–702, June 2011.
- [36] B. Tang, S. Khokhar, and R. Gupta. Turn prediction at generalized intersections. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 1399–1404, June 2015.
- [37] TAPAS. TAPAS Cologne Scenario. https://sourceforge.net/projects/sumo/files/traffic_data/scenarios/TAPASCologne/TAPASCologne-0.24.0.7z/download, 2015. (Visitado 21-11-2021).
- [38] Salomón Torres. Procesamiento de rutas tapas cologne.

<https://github.com/storress/MasterScripts>. (Visitado 21-11-2021).

- [39] Viswacheda Duduku. V, Ali Chekima, Farrah Wong, and Jamal Ahmad Dargham. A study on vehicular ad hoc networks. In *2015 3rd International Conference on Artificial Intelligence, Modelling and Simulation (AIMS)*. IEEE, December 2015.
- [40] Veins. The open source vehicular network simulation framework. <http://veins.car2x.org/>, 2018. (Visitado 21-11-2021).
- [41] Yunpeng Wang, Xuting Duan, Daxin Tian, Guangquan Lu, and Haiyang Yu. Throughput and delay limits of 802.11p and its influence on highway capacity. *Procedia - Social and Behavioral Sciences*, 96:2096–2104, November 2013.
- [42] Yina Wu, Mohamed Abdel-Aty, Ou Zheng, Qing Cai, and Lishengsa Yue. Developing a crash warning system for the bike lane area at intersections with connected vehicle technology. *Transportation Research Record: Journal of the Transportation Research Board*, 2673(4):47–58, April 2019.
- [43] Hailun Zhang and Rui Fu. A hybrid approach for turning intention prediction based on time series forecasting and deep learning. *Sensors*, 20(17):4887, August 2020.
- [44] Alex Zyner, Stewart Worrall, and Eduardo Nebot. Naturalistic driver intention and path prediction using recurrent neural networks. *IEEE Transactions on Intelligent Transportation Systems*, 21(4):1584–1594, April 2020.